Bioinformatic analysis of genomic sequencing data
Read alignment and variant evaluation

Frousios, Kimon

Awarding institution:
King's College London

Bioinformatic analysis of genomic sequencing data: Read alignment and variant evaluation

Kimon Frousios

PhD Thesis

on Bioinformatics Research

Algorithms and Bioinformatics Group

Department of Informatics

School of Natural and Mathematical Sciences

King's College London

December 16, 2014

Preface

I wish to thank my supervisors Prof. C.S. Iliopoulos and Prof. T. Schlitt for their help and guidance during these four years, especially considering the circumstances under which I came to be their student.

I also wish to thank Dr. German Tischler, Dr. Solon Pissis and Dr. Stylianos Arhondakis for introducing me to their respective fields of research and for their collaboration, Dr. M. Simpson, who kindly acted as an unofficial third supervisor and guided me through a big part of my work, as well as Prof. D. Jones and Prof. M.F. Sagot for their constructive criticism and valuable feedback in placing this document in context and making it more understandable.

I would like to also express my gratitude towards the Greek State Scholarships Foundation, who granted me a full scholarship to carry out my studies abroad.

Finally, I wish to thank my friends and loved ones for their encouragement, support and tolerance, with a special nod to Laura, who proof-read this document for me.

Contents

Abstract

The invention and rise in popularity of Next Generation Sequencing technologies has led to a steep increase of sequencing data and the rise of new challenges. This thesis aims to contribute methods for the analysis of NGS data, and focuses on two of the challenges presented by these data.

The first challenge regards the need for NGS reads to be aligned to a reference sequence, as their short length complicates direct assembly. A great number of tools exist that carry out this task quickly and efficiently, yet they all rely on the mere count of mismatches in order to assess alignments, ignoring the knowledge that genome composition and mutation frequencies are biased. Thus, the use of a scoring matrix that incorporates the mutation and composition biases observed among humans was tested with simulated reads. The scoring matrix was implemented and incorporated into the in-house algorithm *REAL*, allowing side-by-side comparison of the performance of the biased model and the mismatch count. The algorithm *REAL* was also used to investigate the applicability of NGS RNA-seq data to the understanding of the relationship between genomic expression and the compartmentalisation of genomic base composition into isochores.

The second challenge regards the evaluation of the variants (SNPs) that are discovered by sequencing. NGS technologies have caused a sharp rise in the rate with which new SNPs are discovered, rendering impossible the experimental validation of each one. Several tools exist that take into account various properties of the genome, the transcripts and the protein products relevant to

the location of a SNP and attempt to predict the SNP's impact. These tools are valuable in screening and prioritising SNPs likely to have a causative association with a genetic disease of interest. Despite the number of individual tools and the diversity of their resources, no attempt had been made to draw a consensus among them. Two consensus approaches were considered, one based on a very simplistic vote majority of the tools considered, and one based on machine learning. Both methods proved to offer highly competitive classification both against the individual tools and against other consensus methods that were published in the meantime.

Chapter 1

Introduction

1.1 Structure of the thesis

This thesis is structured in five chapters:

1. Chapter 1 lays out the structure of the thesis and gives an overall context of the work. In the interest of clarity and continuity, a more detailed background on each subject is given in the corresponding chapters.

2. Chapter 2 addresses the task of mapping NGS reads to a reference genome. Specifically, it assesses an alternative scoring scheme to the one currently widely used by alignment algorithms for NGS reads. This was achieved via a modification of the in-house alignment algorithm *REAL*. Alignment accuracy for simulated reads is presented, for a range of mutation and error rates and it is demonstrated that substitution scores that reflect the evolutionary relationship and mutation biases between two sequences are advantageous even for very short sequences, such as NGS reads.

3. Chapter 3 applies the alignment algorithm *REAL* to the investigation of the connection between the compartmentalisation of genomic base composition into isochores and the expression level at the genomic level. The study is the first to use NGS data for this task and the first to look simul-

taneously at the implication of genomic composition in development and tissue specificity. The results lay to rest the debates around the existence of isochores and the correlation between expression and base composition.

4. Chapter 4 addresses the evaluation of the single–base variants (SNPs) that arise from the analysis of the aligned reads. Two consensus methods are proposed and are shown to perform better than the rival consensus methods and most individual tools.

5. Chapter 5 summarises the conclusions and discusses possible future directions of this work.

1.2 Context of this thesis

Sequencing technology has come a long way since the time when traditional first–generation sequencing techniques required many laboratories around the world to cooperate for years in order to sequence the human genome for the first time. Nowadays, the so-called next generation sequencing (NGS) techniques have reduced the task to a matter of days or hours and the cost has decreased many orders of magnitude [1]. In fact the technology has recently reached the long-anticipated point of being able to re-sequence the whole human genome in as little as 4 hours and at a cost of just $\$1,000$ [2].

As a result, sequencing is progressively replacing a host of older techniques that served as substitutes when sequencing was still slow and expensive, and it's enabling large–scale projects like the 1000 Genomes Projects [3]. The amount of data obtained from a single NGS experiment can be in the order of tens or hundreds of Gigabytes, with high-throughput platforms being able to produce even Terrabytes of data when operated at their full potential. The analysis of sequencing data involves multiple main stages, each of which presents its own challenges, in addition to the overwhelming logistics of storing, managing and analysing the data collected and produced along those stages, which will be introduced in the following subsections.

In more recent years, a third generation of sequencing technologies is emerging, focused on single–molecule sequencing. This development offers a way to bypass the fragment amplification stage and the need for reagent cycles, which are staples of all the sequencing techniques from both the first and second generation.

1.2.1 Sequencing

Three main platforms exist in the field of NGS technologies: 454/Roche, Illumina/Solexa and Life Technologies/Applied Biosystems (ABi). Their main contribution to sequencing was the shrinking of the experimental scale with consequent increase in parallelisation potential, allowing many more fragments to be simultaneously sequenced than was previously possible. All these techniques rely primarily on the initial amplification of the fragments in order to create stronger signal–to–noise ratios.

Both the Illumina and Roche technologies rely on sequencing by synthesis (SBS). Following the anchoring and amplification of the DNA fragments, the four types of nucleotides (dNTPs) are added in repeated cycles in order to enzymatically synthesize a strand complementary to the template fragment. Whenever a dNTP complements the next base in the template, it is added to the synthesized strand and the reaction is detected by optical means. Illumina uses reversibly chain–terminating nucleotides with a different fluorescent dye for each base, allowing only one nucleotide to be incorporated at each cycle, the type of which is determined by the colour of the dye [4]. Roche, on the contrary, does not use chain-terminating nucleotides. Instead, the sequence is determined by adding nucleotides in turns, while incorporation is detected by means of light emitted when pyrophosphate (a by-product of strand extension) is used up by luciferase [5]. This allows homopolymer runs to be synthesized in one step, with a respective increase in the intensity of light emitted. The main selling point for the Roche platforms was their ability to generate much longer reads (initially 250bp, later 500–800bp) than its rivals (Illumina: initially 32bp,

now up to 300bp, Solid: $< 100bp$). However, the improvements in read length and throughput of rival platforms have made pyrosequencing less cost-effective and it is scheduled to be discontinued [2]. Another SBS platform, similar to pyrosequencing, is the Ion Torrent by Life Technologies [6]. There, nucleotides are added in turns, but instead of using optical systems, strand extension is detected by the change in pH caused by the release of protons. The platform is capable of 100-400bp long reads and is soon set to quadruple its throughput capacity to the point of sequencing the whole human genome to 40-fold coverage in 4 hours [2].

An older Life Technologies platform, the ABi SOLiD, is different in mechanism and interpretation. The anchored and amplified fragments are read by ligation of complementary oligonucleotides [7]. The first two bases of each oligo are the important ones, creating a set of 16 oligos. Four fluorescent dyes are used, each for four of the oligos, and the sequence is deduced by means of the overlap between the oligos. Because each base is read twice, the method is considerably more accurate at base–calling than the SBS methods, however, the maximum length of the reads obtained is low compared to the latest versions of other platforms.

These technologies suffer from the need to amplify the fragments prior to sequencing, so as to increase the signal strength over the background noise. This is prone to causing miscall errors, when the theoretically identical clones of a fragment disagree on the type of a base. This can be due either to erroneous bases inserted by the polymerases or due to the clones falling out of synchronisation with one another. The severity of the former depends on how early in the amplification cycles the error was inserted, the severity of the latter progressively increases as the reads become longer. Additionally, the Roche and Ion Torrent platforms are prone to misjudging the length of homopolymer runs, because the difference in signal intensity between longer homopolymers becomes smaller [6]. Lastly, amplification and ligation are biased in terms of the size and GC composition of the fragments [8].

In an effort to bypass the problems of amplification, upcoming third–generation technologies focus on single–molecule sequencing (SMS). The Helicos Genetic Analysis System [9] was a platform that applied sequencing by synthesis to single molecules with fluorescent reversibly chain–terminating nucleotides, similarly to Illumina, but using a single polymerase molecule. The sensitivity and geometry of the optics was optimised so as to ensure low background noise but, in practice, the signal detection remained weak and resulted in many bases not being read, causing artificial deletions in the read sequence [10]. The Helicos platform is no longer available, but the intellectual property is being acquired by Illumina and other rivals.

Another SMS platform, from Pacific Biosciences, also employs SBS with differentially labelled nucleotides, but differs from Illumina and Helicos in that it employs single polymerases anchored in wells together with powerful high–speed optics to detect the four–coloured fluorescence in real time using confocal microscopy [11,12]. Once a nucleotide is incorporated and the colour is recorded, the dye is cleaved and diffuses away from the focus area. Furthermore, the templates are prepared in a way that they become circular, allowing the same region to be read multiple times. Read length is only limited by the eventual polymerase damage and can reach from $1kbp$ to $40kbp$, greatly surpassing the available rival technologies [2].

Finally, the future of sequencing appears to be nanopore sequencing. No amplification, labels or synthesis are required; instead the molecule itself is read directly as it passes through a pore separating two compartments. The principle is based on each building block of the molecule having a different signature effect on the electrical potential of the membrane containing the pore [13–16]. Oxford Nanopore Technologies has already released the first such devices, including an extremely portable device that is palm–sized and requires minimal sample preparation[1]. It also has the potential of analysing more than just DNA in the sample. The implications of having such portable, versatile and easily applicable

[1] http://nanoporetech.com

analytical technology available are far–reaching, with diagnostic medicine and environmental surveys being two of the fields likely to benefit greatly from such technology.

1.2.2 Base calling

Base calling is the process of inferring the read sequence from the raw signals of the sequencer. For NGS platforms this generally means converting the light intensities and colour emitted by the fluorescent nucleotides as they are incorporated into the synthesized or ligated strand. As mentioned in the previous subsection, errors can be made in this stage due to flaws of the sequencing processes. Firstly, the polymerases used in amplification occasionally incorporate the wrong base. When such errors occur in an early cycle, they propagate to a large number of clones and cause a high noise background. Secondly, the clones eventually start falling out of phase with each other and are no longer read all at the same position (phasing or desynchronisation), creating background noise that becomes higher as sequencing progresses. Thirdly, the different fluorescent dyes often interact due to overlap of their absorption and emission spectra (cross–talk) [17, 18]. All these errors affect the read sequence and, in turn, this can alter the optimal mapping or assembly location of the read, or cause the misinterpretation of errors as true sequence variants.

Conversion of the emitted light intensities into sequence is achieved using software, including proprietary and third–party options. Among the latter, Phred [19, 20], initially released for Sanger sequencing, has set the standard and output format for the read sequence and the encoding for the respective quality of each base in the sequence and that standard is also widely followed by NGS platforms today. Phred and programs based on its strategy try to identify the single most likely sequence for the read and this version of the sequence is then passed on downstream for alignment or assembly, discarding the information of sub-optimal base calls in the process. Contrary to this, Rolexa [18] also encodes suboptimal base qualities and Slider [21] takes it a step further by fusing the

calling stage with the mapping stage, so as to consider the probabilities of the sub-optimal calls in cases of alignment mismatches. As a side–effect it also facilitates logistics by skipping the intermediate files used to store the read sequences. Looking at suboptimal calls is a good way to address low–quality base calls caused by polymerase errors, however, it is not likely to influence the alignment of low quality–bases that were caused by desynchronisation, as the suboptimal calls in this case come from a different position along the sequence and are thus meaningless in Slider's context.

Several methods have been published that address the problem of desynchronisation and dye cross-talk. The first alternative to Illumina's proprietary base–caller, Bustard, was Alta-Cyclic [17] which relies on machine learning techniques. However, the method is cumbersome and costly, as it requires a flowcell to be sacrificed for the sequencing of a well–known control sequence, which would then be called using Bustard, so as to use the result as a training set for the model. A number of other base–calling algorithms have been released, some of which use explicit probabilistic [18,22–25] or other mathematical [26–28] models of the sequencing chemistry and its errors, while others employ machine learning techniques [17,29,30]. The increased accuracy of these methods comes from accounting for more variables than the proprietary software, but the trade–off is the increase in computational load and the reduction of speed [28,31]. Machine learning appears to be the fastest solution, though the mathematical models can be sped up if various assumptions are made about the uniformity and homogeneity of the sequencing process across clusters, cycles, flowcells or separate runs [23,27].

1.2.3 Read mapping or assembly

As presented in 1.2.1, modern NGS platforms are capable of producing anywhere from several Gigabytes to a few Terrabytes of data per run, in the form of tens or hundreds of millions of reads, and sometimes even billions of reads, whose lengths nowadays vary between 100–800bp, depending on the platform. This

thesis focuses on data obtained with Illumina platforms, whose typically read size now is 100–300*bp*. However, when this work began, reads of up to 100*bp* were still uncommon and subject to high error rates, so the context of the work presented in chapter 2 at the time was reads in the order of 30–70*bp*. The findings of the work can be applied to reads of all lengths, though their impact might be less pronounced.

Typically, NGS platforms are used for genome re-sequencing, rather than *de novo* sequencing of an unknown genome. This means that the reads obtained from the platform are then aligned to the previously known reference sequence for the organism in question. For large genomes like the human one, the sheer number of reads surpasses the processing and storage capacity of common computers, especially using traditional alignment algorithms like dynamic programming [32,33] and heuristic approaches [34,35], both of which are proven to be too slow for this task [36]. NGS, thus, has created the need for extremely efficient algorithms that carry out the read mapping in a more reasonable time while maintaining the memory and processor requirements accessible without the need of supercomputers and scalable to the ever–increasing dataset sizes and read lengths.

Additional challenges for this stage are the frequent natural presence of exact and approximate repeats in the genome, which causes reads to align similarly well to multiple locations. The situation is exacerbated by the presence of natural variability and sequencing errors, both of which can alter the optimal mapping position of a read. The effect of variants and errors can be alleviated with longer reads, as longer sequences are statistically more unique. Repeats are more difficult to address, as they can be much larger than the reads. Mate–pair reads can solve some repeats by means of the large fixed-length insert sequence that places the two reads of the pair at a fixed distance from one another.

Many read mapping programs have been published to address these issues [21, 36–52], including *REAL* [53, 54], an efficient read aligner developed within the Algorithms and Bioinformatics Group at King's College London.

The solutions focus mainly on algorithm efficiency and evaluation of the quality of alignments. Repeats generally cannot be solved unless the reads or insert sizes are comparable to or larger than the size of the repeats, thus only platform improvements in read and insert length can solve this problem. In this aspect, third generation platforms are very promising.

Of course, re-sequencing is limiting NGS to genomes that are already known, while novel genomes are generally sequenced by the slower and more expensive traditional techniques that produce reads long enough to disambiguate many of the repeats. However, considering the plummeting costs of NGS, it is desirable to apply NGS to *de novo* sequencing but, without the presence of a reference sequence, reads have to be assembled without help. Traditionally this was done by finding the overlaps among reads [55], however this approach does not scale well for large numbers of very short reads. A small number of algorithms have been proposed [56–62], that rely on finding Euler paths to traverse de Bruijn graph representations of the reads [63]. NGS assembly algorithms suffer from all the same challenges as NGS mapping algorithms and are known to create smaller contiguous sequences than traditional Sanger data and cannot reliably extend these contigs through repetitive regions [64]. Although *de novo* assembly is a more powerful tool than read mapping, mapping is preferred when possible, as the reference sequence provides context and structure to the mapped reads, compared to the disassociated contigs resulting from assembly. Of course, the increasing read lengths obtained with more modern versions of the platforms are as beneficial to assembly as they are to mapping and make NGS a more viable option for *de novo* sequencing tasks.

The REAL algorithm

REAL [53, 54] is a fast and simple read alignment algorithm for NGS reads, designed within the Algorithms and Bioinformatics group at King's College London. It guarantees to find all gap–less alignments with up to a given number of mismatching bases between the read and the reference and reports the

alignment with the fewest mismatches. In case of a tie, it offers the options to report either none of the conflicting alignments or all of them.

A suffix of given length is taken from each read. This suffix is allowed to have up to a given number of mismatches in an alignment (by default 2), separate from the maximum number of mismatches allowed for the entire read (by default 5). This makes it possible to split the suffix into fragments (by default 4) such that some fragments are guaranteed to have no mismatches (in the default case 2 fragments), thanks to the pigeon-hole principle. By searching for exact matches of all the pairwise combinations of these four fragments into an index of the reference that has been pre–processed in the same way, it is possible to locate all the candidate alignments with 2 or less mismatches, as well as some with more. These are then scanned to ensure that neither the suffix nor the entire read have more mismatches than the respective allowed limits. Then the alignment with the least mismatches is reported.

The algorithm has comparable speed and yield to SOAP2 and Bowtie, two of the fastest and most popular non-commercial read alignment tools available at the time. Both these tools require the reference to be pre-processed and indexed using the Burrows–Wheeler Transform. *REAL*, in contrast, does not store a pre-processed index of the reference, yet is able to perform the alignments in similar time as the tools with the pre-compiled index. *REAL*, however, has several drawbacks compared to other tools. It can only align single–end reads, without gaps, and offers no other useful features such as tag recognition and removal. An algorithm to incorporate gaps into the alignments has been developed [65], but has not yet been incorporated into the programme.

1.2.4 Variant calling

Following the alignment or the direct assembly of the reads, the result is then assessed for the presence of known and new variants. This includes the identification of single base polymorphisms (SNPs), copy number variations, insertions and deletions, usually with the aim to uncover any underlying genetic mark-

ers that are associated with susceptibility, progress or cause of a disease, but also with the aim to expand our knowledge and understanding of the extent of natural variation.

The process involves comparison of the new data against the reference sequence and variation databases and is more complex than it first appears. The quality of the bases must be taken into consideration and conflicts between overlapping reads must be resolved. Such conflicts may arise from sequencing errors, misaligned reads, contamination by foreign DNA, or by the genuine presence of two or more versions of a locus as a result of diploidism and polyploidism. The task often involves re-alignment of reads based on the context of their overlapping other reads and recalibration of quality values. A number of software tools have been developed for this task, with the Genome Analysis Toolkit (GATK) [66], SAMtools [67] and Atlas2 [68] being the most widely used and accurate [69], though others also exist [42, 70–74].

Most importantly, the reliability of variant calling is directly dependent on read coverage, that is the number of overlapping reads that cover each base of the sequence in question, as is demonstrated by the low level of agreement among the different tools when the read coverage is low [75]. Common practice is, thus, to aim for high coverage, so that sequencing errors and chance misalignments can be more reliably separated from the genuine sequence by means of overwhelming consensus among the reads.

1.2.5 Interpretation

Knowing the variants in each particular genome enriches our knowledge of a species' genetic diversity, but on its own it offers very little to our understanding of this diversity. Therefore it is important to associate these variants to specific changes in the phenotype of the organisms bearing them. Indeed, much of the data generated is aimed at the discovery of disease–causing genetic variation. Thus the next key challenge in NGS data analysis is the interpretation of the functional consequences of variant alleles. However, given the extensive

number of variants identified by whole genome or exome re-sequencing studies, it is infeasible to experimentally interrogate the functional consequences of all variant alleles at all gene loci.

A number of bioinformatics solutions for the annotation, scoring and classification of variants have been developed to address this challenge [76–102]. Such tools are providing a supportive role in the experimental validation of disease–related alleles, by prioritising candidate variants with predicted functional consequences as causes of specific inherited diseases and cancers. These bioinformatics approaches draw from a broad range of existing knowledge about the structure, function and conservation of genes, transcripts and proteins in which the variants are located, as will be presented in more detail in chapter 4.

Although, the study of variation in general and its connection to various pathological conditions in particular are a major area of research, there are other applications of NGS technologies. One that is especially interesting is the study of gene expression [103]. Until now, microarrays have been the primary means to study expression, but they have technical limitations that NGS does not. These are the limited number of wells and the limited number of known genes, the need for prior knowledge of the sequence so as to create appropriate hybridization probes, and limited dynamic range leading to saturation phenomena, whereby it is impossible to detect differences between higher levels of expression once all available probes are hybridized. In contrast, NGS platforms make no *a priori* assumptions about the studied sequences and are, thus, well–suited for the discovery of rare or unknown transcripts, the study of alternative splicing and the detection of pathological irregularities like gene–fusion. An application of NGS to study transcription will be presented in chapter 3. Finally, the applications of NGS keep broadening and replacing older more complex and less efficient methods. Aside from revolutionizing genotyping and transcriptomics for research and diagnostic purposes, NGS is also used in the study of epigenetics, such as methylation patterns and chromatin structure, including nucleosome positioning, DNA accessibility and histone modifications [104].

Chapter 2

Alignment of short reads using evolutionary scores

2.1 Background

NGS technologies produce relatively short sequence reads, from which the original sequence must be inferred. This can be achieved either by relying on the overlap between reads in order to deduce the original sequence, or by using a previously sequenced and assembled similar sequence as a guide. The latter is similar to solving a jigsaw puzzle when the final picture is known in advance, the former is similar to solving the puzzle blindly. Both options present challenges.

Unassisted (*de novo*) assembly is hindered by the very limited alphabet that genomes are composed of (only four bases), the very short length of reads produced by certain NGS platforms (from $27bp$ at the time this work began), the presence of extensive repeats in the genomes and the presence of sequencing errors in the reads, all of which contribute to rendering the unambiguous assembly of the reads into long contiguous sequences very complicated [105]. Assisted assembly using a pre-existing and nearly identical sequence as a guide depends on the availability of a suitable such reference sequence and is liable to the same

problems as unassisted assembly. However, the *a priori* knowledge offered by the reference sequence simplifies the assembly process and helps obtain a more structured and more readily interpretable overview of the sequenced regions and of the number and location of areas of ambiguity or anomaly.

In this chapter, the focus is on the alignment of short reads using a pre-existing sequence as reference, a process commonly known as read mapping. The sheer number and short length of the reads differentiates it from the classical problems of sequence alignment and database searching, because the well-established dynamic programming [32, 33] and heuristic [34, 35] algorithms are proven to be too slow for this task [36]. A great number of tools have been developed to specifically address the task of read mapping [21, 36–52], including the in-house algorithm *REAL* [53, 54]. All of these tools identify the best alignment location for each read based on the least number of mismatching bases between the read and the reference.

Although the least number of mismatches presents a simple and mathematically well formulated solution, it does not guarantee to find the biologically most relevant alignment. One cause of this is the fact that the sequenced genome will differ from the reference one by a small but not negligible percentage of bases, as part of the natural diversity among individuals. The other cause of this is the presence of sequencing errors. Distinguishing sequencing errors from genuine variations is possible with the use of the probability of error at each position of a read, as recorded by the sequencing machine. Various models have been proposed to incorporate quality information into the alignment score [42, 52, 106, 107].

With sequencing errors accounted for, a number of reads may still fail to be unambiguously mapped, as genuine variation can also lead to alternative alignment locations with the same number of mismatches. In these cases a choice must be made to discard all the alignments of the read, to choose one alignment at random or to keep all of the alignments. None of these solutions is ideal and this chapter aims to investigate whether the incorporation of likelihood ratio

scores (log odds scores) [108] for the substitutions can help disambiguate such cases in the alignment of reads from the human genome. A previous attempt to incorporate base quality values with a substitution matrix [52] was based on a scoring formula that is not justified [107]. A corrected model has been successfully applied in the cross-species alignment of reads from insects [107].

Given a model by which to combine a substitution matrix with the base quality information recorded by the sequencing machine, the next step is to choose a suitable substitution matrix. The commonly used matrices [109, 110] are general-purpose matrices available at various levels of sequence similarity. However, detailed specific information on the variation between human individuals is available [111], enabling the creation of a matrix tailored to humans. This information highlights two types of bias in the observed frequencies of substitutions: The bases G and C are found to be substituted much more often than A and T, and the transition type of substitutions ($A \leftrightarrow G$ and $T \leftrightarrow C$) is much more frequent than the others.

In this chapter, the substitution matrix for humans is calculated based on the GC bias, the transition bias and the base composition bias of the human genome. This matrix is implemented and incorporated into the in-house read alignment algorithm *REAL* and tested on simulated reads.

2.2 Methods

The aim of this chapter is to create a scoring matrix for nucleotide substitutions, with which to discriminate between genuine genetic diversity and artefacts caused by sequencing errors during the alignment of reads to a reference. To this end, the classic likelihood ratio formula will be used [109, 110, 112]. This requires knowledge of two parameters:

- the probability of two specific bases at specific locations being aligned to each other by pure chance, and
- the probability of them being aligned because of a genuine evolutionary

relationship (homology) between the sequences.

These can be calculated for a given level of sequence identity (or equivalently a given mutation rate) and a given base composition for each sequence. However, the mutation rate is under the influence of multiple biases that affect the relative probabilities of the various substitution possibilities. In order to make the scoring formula flexible and applicable to different organisms, it is useful to formulate the substitution probabilities in relation to the biases, so that the biases can be passed directly as arguments to the algorithm, without imposing the inference of the probabilities on the user.

In the following sections, this matrix will be constructed and then tested using simulated reads. In order to simulate reads from a given reference sequence, it is necessary to have a formula with which to simulate genetic variability, as well as a formula with which to introduce errors.

2.2.1 Preliminaries

Before proceeding, it is necessary to define the symbol conventions used in this chapter, as well as some very basic relationships between them.

Definition 2.1.

- *Let* $\mathbb{L} = \{A, C, G, T\}$ *be the set of bases from which the reference sequence is built.*
- *Let* $\mathbb{L}' = \{A', C', G', T'\}$ *be the set of bases from which the reads are built.*

Definition 2.2.

- *The probability (frequency) of base* $X \in \mathbb{L}$ *in a given sequence will be noted as* $p(X)$*. Of course:* $p(A) + p(C) + p(G) + p(T) = 1$.
- *The overall probability (frequency) of* $X \in \mathbb{L}$ *in the reference being replaced by* $Y \in \mathbb{L}'$ *in the reads will be noted as* $p(Y \cap X)$*.*

- *The probability of a given base $X \in \mathbb{L}$ in the reference being substituted by $Y \in \mathbb{L}'$ in a read will be noted as $p(Y|X)$ (probability of a given base in a read being Y, given that the aligned reference base is X).*

The observed frequency of substitution of $X \in \mathbb{L}$ by $Y \in \mathbb{L}'$ is proportional to the frequency of X and the probability that X is substituted by Y (as dictated by the multiplication axiom in Probability Theory):

$$p(Y \cap X) = p(X) \times p(Y|X) \tag{2.1}$$

Definition 2.3. *As a consequence of base complementarity and the double–stranded nature of DNA, it is often necessary to refer to the joint frequency of A and T or G and C in a sequence:*

- $p(A \cup T) = p(A) + p(T) = 2 \times p(A) = 2 \times p(T)$
- $p(G \cup C) = p(G) + p(C) = 2 \times p(G) = 2 \times p(C)$

Definition 2.4. *Let $M \in \mathbb{L}'$ symbolize a base in the read that differs from the base in the reference. Then $p(M)$ is the overall probability (frequency) of substitution. The are two types of substitution:*

- *Transition (M_{ts}) — When a purine (A,G) or pyrimidine (T,C) is substituted by the other purine or pyrimidine respectively:*
  $M_{ts} = (A' \cap G) \cup (G' \cap A) \cup (T' \cap C) \cup (C' \cap T)$
  $p(M_{ts}) = p(A' \cap G) + p(G' \cap A) + p(T' \cap C) + p(C' \cap T)$
- *Transversion (M_{tv}) — When a purine is substituted by a pyrimidine or vice versa:*
  $M_{tv} = (C' \cap A) \cup (T' \cap A) \cup (A' \cap C) \cup (G' \cap C)$
  $\cup (C' \cap G) \cup (T' \cap G) \cup (A' \cap T) \cup (G' \cap T)$
  $p(M_{tv}) = p(C' \cap A) + p(T' \cap A) + p(A' \cap C) + p(G' \cap C)$
  $+ p(C' \cap G) + p(T' \cap G) + p(A' \cap T) + p(G' \cap T)$

By definition: $p(M_{ts}) + p(M_{tv}) = p(M)$.

2.2.2 Calculation of the substitution probabilities

The various substitution frequencies within a given species can be measured, and such information is available for humans [111]. Using these measurements along with intrinsic properties of DNA, the actual substitution probabilities can be inferred.

For generality and simplicity towards a user of an alignment algorithm in which this matrix would be applied, it is useful reduce the number of parameters by expressing the substitution probabilities relatively to a handful of parameters, namely the overall substitution rate and the substitution biases. Up to this point, the only bias considered has been the composition of the reference, which affects the relative abundance of the bases. Two more biases have been measured in humans [111]: i) The substitution of G or C is more frequent than the substitution of A or T, despite G and C being scarcer in the human genome, and ii) the four transitions are more frequent than the eight transversions.

The GC mutability bias

Definition 2.5. *Let B be the ratio of the observed mutability of G and C compared to A and T:*

$$B = \frac{p(M \cap G) + p(M \cap C)}{p(M \cap A) + p(M \cap T)}$$

If the probability is the same for each type of substitution (no bias), then the ratio of mutations should be the same as the ratio of the base occurrence (as dictated by Definition 2.3 and Equation 2.1):

$$B_{neut} = \frac{p(G \cup C)}{p(A \cup T)} \tag{2.2}$$

In the case of humans, $B = 2$ [111]. Furthermore, this bias extends to individual substitution types:

$$B = \frac{p(A' \cap G)}{p(G' \cap A)} = \frac{p(T' \cap C)}{p(C' \cap T)} = \frac{p(A' \cap C)}{p(C' \cap A)} = \frac{p(T' \cap G)}{p(G' \cap T)} = \frac{p(C' \cap G)}{p(T' \cap A)} = \frac{p(G' \cap C)}{p(A' \cap T)} \tag{2.3}$$

This bias splits the transitions and transversions into two subgroups each, based on the original base [111]:

$$p(A' \cap C) = p(G' \cap C) = p(T' \cap G) = p(C' \cap G) \tag{2.4}$$

$$p(C' \cap A) = p(G' \cap T) = p(A' \cap T) = p(T' \cap A)$$

$$p(G' \cap A) = p(C' \cap T) \tag{2.5}$$

$$p(A' \cap G) = p(T' \cap C)$$

The transition bias

Although transitions represent only one third of the possible substitutions, they are observed considerably more frequently than transversions. These frequencies are measured as fractions of the total number of mutations. Transitions and transversions can be considered composite events, consisting of the event of a substitution occurring and the event of that substitution turning out to be either a transition or a transversion.

Definition 2.6. *Let $p(M_{ts}|M)$ and $p(M_{tv}|M)$ be the probabilities of a mutation turning out to be a transition or a transversion respectively (probability of transition or transversion, given that a substitution took place). Then, the multiplication axiom gives:*

- $p(M_{ts}|M) = \dfrac{p(M_{ts})}{p(M)}$
- $p(M_{tv}|M) = \dfrac{p(M_{tv})}{p(M)}$

By definition: $p(M_{ts}|M) + p(M_{tv}|M) = 1$.

In the case of humans, $p(M_{ts}|M) = 0,71$ and $p(M_{tv}|M) = 0.29$ [111].

The substitution probabilities

Using the inherent properties of DNA base substitutions and the mutational biases described in the previous subsections, it is now possible to express the substitution probabilities as functions of the GC mutability bias B, the transition bias $p(M_{ts}|M)$, the substitution rate $p(M)$ and the composition of the reference $p(G \cup C)$.

Theorem 2.1. *The probability of each transition type is given by the following equations:*

- $p(G'|A) = p(C'|T) = \dfrac{1}{B+1} \times \dfrac{p(M_{ts}|M) \times p(M)}{1 - p(G \cup C)}$
- $p(A'|G) = p(T'|C) = \dfrac{B}{B+1} \times \dfrac{p(M_{ts}|M) \times p(M)}{p(G \cup C)}$

Proof. Starting with the definition of transitions (Definition 2.4):

$p(M_{ts}) = p(A' \cap G) + p(G' \cap A) + p(T' \cap C) + (C' \cap T)$

Using Equations 2.3 and 2.5:

$\Rightarrow p(M_{ts}) = 2 \times p(A' \cap G) + 2 \times p(G' \cap A)$

$\Rightarrow p(M_{ts}) = 2 \times B \times p(G' \cap A) + 2 \times p(G' \cap A)$

$\Rightarrow p(M_{ts}) = 2 \times (B+1) \times p(G' \cap A)$

Using to Definition 2.6 and Equation 2.1:

$\Rightarrow p(M_{ts}|M) \times p(M) = 2 \times (B+1) \times p(A) \times p(G'|A)$

Using to Definition 2.3:

$\Rightarrow p(M_{ts}|M) \times p(M) = 2 \times (B+1) \times \frac{1-p(G \cup C)}{2} \times p(G'|A)$

$\Rightarrow p(G'|A) = \frac{1}{B+1} \times \frac{p(M_{ts}|M) \times p(M)}{1-p(G \cup C)}$

Similarly for $p(C'|T)$, $p(A'|G)$ and $p(T'|C)$. □

Theorem 2.2. *The probability of each transversion type is given by the following equations:*

- $p(C'|A) = p(T'|A) = p(A'|T) = p(G'|T) = \dfrac{1}{2(B+1)} \times \dfrac{\big(1 - p(S|M)\big) \times p(M)}{1 - p(G \cup C)}$
- $p(C'|G) = p(T'|G) = p(A'|C) = p(G'|C) = \dfrac{B}{2(B+1)} \times \dfrac{\big(1 - p(S|M)\big) \times p(M)}{p(G \cup C)}$

Proof. Similarly to the proof of Theorem 2.1, but starting with the definition of transversions instead (Definition 2.4) and using Equation 2.4 instead. □

Theorem 2.3. *The probability that the base in the read remains the same as in the reference is:*

$$p(X'|X) = 1 - p(Y_1|X) - p(Y_2|X) - p(Y_3|X)$$

where $X \in \mathbb{L}$ is the base in the reference and $X', Y_1, Y_2, Y_3 \in \mathbb{L}'$ are the possible bases in the read, such that $X = X'$ and $X \neq Y_1 \neq Y_2 \neq Y_3$.

Proof. The probability of a base to remain unchanged, and the probability for it to be substituted by each of the three other bases cover the entire sample space of outcomes for that base and therefore add up to 1. □

2.2.3 Simulation of errors and base qualities

Two main types of error can occur in sequencing on Illumina or similar platforms:

- The first type occurs when the clones disagree on the base being read, thus emitting an ambiguous signal. This can be caused either by the introduction of the wrong base by the polymerases during the amplification stage, or by the progressive de-synchronization of the clones as sequencing proceeds. In both cases, the probability of error (P_Q) is recorded and encoded in the base call quality Q.
- The other type concerns any random error that might occur at any stage of handling, such that it would not be encoded in the base qualities. This error rate (P_c) will be assumed to be constant throughout the length of all reads.

Definition 2.7. *The total probability of error at a position x is the sum of the*

constant error probability and the base call error probability:

$$P_{err}(x) = P_Q(x) + P_c$$

Base call errors

Definition 2.8. *The Phred-like base quality Q is by definition related to the probability of error P_Q as follows [20]:*

$$Q = -10 \times log_{10}(P_Q)$$

In order to simulate base qualities for a simulated read, it is necessary to have a function with which to relate the error probability to the position x in the read. Upon observation of a number of base quality profiles from various sequencing runs on an Illumina GA platform (Dr. M.Simpson, private communication), it became apparent that the error probability along a read dropped in a non-random way. Specifically, it was observed that, despite the irregularities and differences between the various sequencing runs, a linear trend appeared to roughly fit the co-variation between the inverse of the error probability and the position on the read (Fig. 2.1):

$$\frac{1}{P_Q} = \alpha \times x + \beta \iff P_Q = \frac{1}{\alpha \times x + \beta} \tag{2.6}$$

The constants α and β of the above linear function can be determined if two points of the function are known. Two such points indeed exist:

- The highest quality base is typically the first base of the read, and modern practice arbitrarily caps base call qualities at $Q = 40$, reserving higher qualities for multiple alignments.
- In a typical sequencing run the quality progressively drops as the clones fall out of synchronization until it reaches $Q = 0$, and consequently the error probability becomes $P_Q = 1$. It is possible for the quality to drop

drastically halfway along a read and then come up again, as a result of a temporary random external disruption during sequencing, but this is an anomaly.

Theorem 2.4. *The probability of error (P_Q) at position x of a simulated read is given by the following equation, where X_D is the desynchronization length (at which $Q = 0$ and $P_Q = 1$):*

$$P_Q(x) = \begin{cases} \dfrac{1}{10^4 \times \left(1 - \frac{x}{X_D}\right)} & \text{for } x < X_D \text{ and } x, X_D \in \mathbb{N} \\ 1 & \text{for } x \geq X_D \text{ and } x, X_D \in \mathbb{N} \end{cases}$$

Proof. The definition of the parameters provides two fixed points, which can be used to determine the values of the constants α and β in Equation 2.6:

- At position $x = 0$: $Q = 40$ (by convention), so $P_Q(0) = 10^{-4}$.
  Equation 2.6 $\Rightarrow 10^{-4} = \frac{1}{\alpha \times 0 + \beta} \Rightarrow \beta = 10^4$.
- At position $x = X_D$: $Q = 0$ (by definition), so $P_Q(X_D) = 1$.
  Equation 2.6 $\Rightarrow 1 = \frac{1}{\alpha \times X_D + 10^4} \Rightarrow \alpha = lim_{x \to X_D}(-\frac{10^4}{x}) \approx -\frac{10^4}{X_D}$.

The illegal division by 0 at position $x = X_D$ is avoided by explicitly specifying that $P_Q(X_D) = 1$, which is the definition of X_D in the first place. Substituting the values of α and β in Equation 2.6 concludes this proof. □

2.2.4 Implementation of read simulation

To simulate reads, a script in the Perl language has been written. The input consists of one contiguous reference sequence in FASTA format, with run-time parameters determining the desired sequence identity level $\big(1 - p(M)\big)$, the observed fraction of mutations that are transitions ($p(M_{ts}|M)$), the bias of substitutions originating from G or C (B), the desired read length, the desired sampling interval, the desired desynchronization length (X_D) and the constant error rate (P_c). The base composition ($p(G \cup C)$) of the reference sequence is

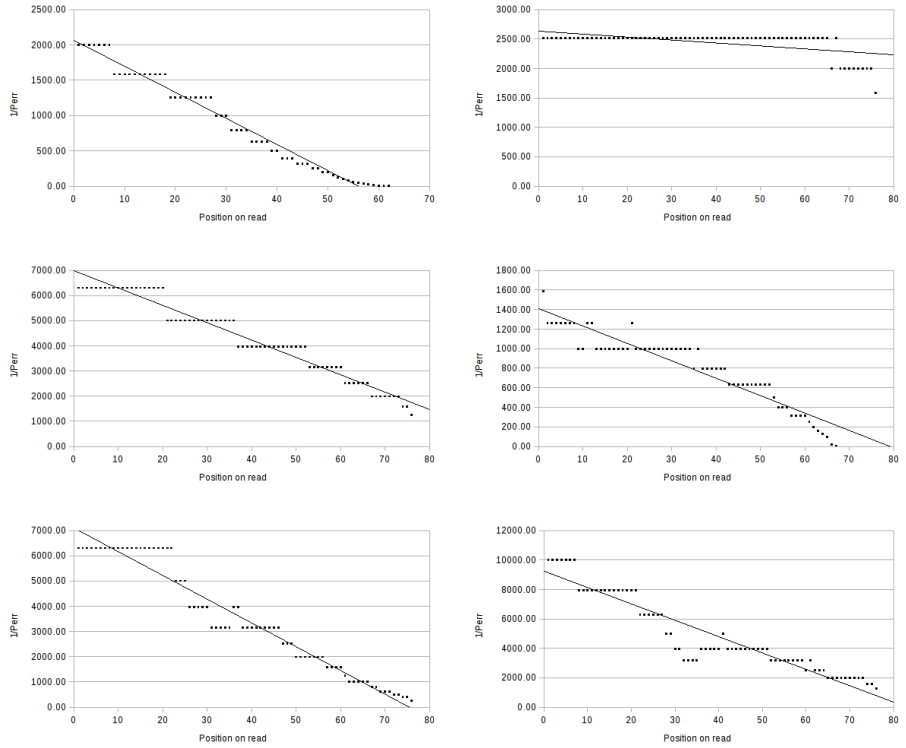


Figure 2.1: Co-variation plots between the position in a read and the inverse of the median error probability for reads of a few typical sequencing runs.

measured at run-time. All these entities and symbols have been explained in the previous sections (2.2.1, 2.2.2 and 2.2.3).

Substitution simulation

The substitutions are applied to the reference sequence prior to sampling, so as to simulate the real-life process and ensure that overlapping reads present the same substitutions. It also allows the mutated sequence to be stored in a new file for future re-use.

Let $T = t_1 t_2 \dots t_n$ be the reference string of length n and consisting of bases $t_i \in \mathbb{L}$, and let $T' = t'_1 t'_2 \dots t'_n$ be the mutated reference consisting of bases $t'_i \in \mathbb{L}'$. The probability of each outcome (Section 2.2.2) is stacked and compared against a randomised value, in order to determine if a base should be mutated and which of the other three bases it should be mutated into (Algorithm 1).

Algorithm 1 Generation of substitutions.

for `i` $\leftarrow 1$ **to** `referenceLength` **do**
  if $t_i = A$ **then**
    $r \leftarrow rand(), \quad 0 \leq r \leq 1$
    if $r < p(C'|A)$ **then**
      $t'_i \leftarrow C$
    else if $r < p(G'|A) + p(C'|A)$ **then**
      $t'_i \leftarrow G$
    else if $r < p(T'|A) + p(G'|A) + p(C'|A)$ **then**
      $t'_i \leftarrow T$
    else
      $t'_i \leftarrow t_i$
    end if
  else if $t_i = C$ **then**
    ...
  else if $t_i = G$ **then**
    ...
  else if $t_i = T$ **then**
    ...
  end if
  `i` $\leftarrow$ `i` $+ 1$
end for

Read generation

Reads are sampled from the mutated reference T' in a non-random way. Sampling begins at one end of the reference and reads are sampled from the reference at a regular interval. Every second read is then replaced with its reverse complement, in order to simulate reads being created from both the forward and the reverse strand of the reference. The number of reads generated depends on the sampling interval specified. Coverage of the reference is determined by the length of the reads and the sampling interval. Any reads found to contain unknown nucleotides ('N') are discarded (Algorithm 2).

Algorithm 2 Generation of reads.

```
for position ← 1 to textLength do
  simulatedRead ← substring(T′, position, readLength)
  if remainder(position/2) = 0 then
    simulatedRead ← reversecomplement(simulatedRead)
  end if
  position ← position + samplingStep
end for
```

The final stage is to generate random errors in the reads. The user–controllable parameters are the length at which de-synchronization occurs (X_D), which allows the calculation of the base call error (P_Q), and the value of the constant error rate (P_c). Application of only one of the two types of errors is possible via the use of appropriate parameter values. When the de-synchronization length is set to a number much larger than the read length, the base qualities will remain high throughout the read and practically only the constant error rate will be in effect. Conversely, specifying a very low value for the constant error rate allows the generation of only position-dependent errors. The errors are generated by comparing a randomised value against the total error probability per position in order to determine whether a base should be changed. If a change should occur, the three possible outcomes have equal probabilities (Algorithm 3).

The simulated reads are output in FASTQ format. The coordinates (counting from 0) and the identity of the reference where each read is sampled from are preserved in the read's title field, to enable the quantification of correctly

Algorithm 3 Generation of errors.

for $i \leftarrow 1$ **to** `readLength` **do**
  $r \leftarrow rand(), \quad 0 \leq r \leq 1$
  if $r < P_{err}(i)$ **then**
    $r \leftarrow rand(), \quad 0 \leq r \leq 3$
    if $t_i = A$ **then**
      if $r < 1$ **then**
        $t'_i \leftarrow C$
      else if $r < 2$ **then**
        $t'_i \leftarrow G$
      else
        $t'_i \leftarrow T$
      end if
    else if $t_i = C$ **then**
      ...
    else if $t_i = G$ **then**
      ...
    else
      ...
    end if
  else
    $t'_i \leftarrow t_i$
  end if
  `i` $\leftarrow$ `i` $+ 1$
end for

and wrongly aligned reads.

2.2.5 Alignment scoring

To map the simulated reads to the reference sequence, the fast in-house algorithm REAL [54] served as the foundation. Originally, REAL simply used the number of mismatching bases to determine which alignment was the most likely. In order to allow different mismatch types to be weighted differently, REAL was extended to implement a different score for each mismatch type, following a process similar to that used for the design of the PAM and BLOSSUM substitution matrices [109, 110].

Creation of the scoring matrix

Scoring matrices are typically based on the logarithms of the odds ratios between the probabilities of the bases being aligned due to homology and the probabilities of the bases being aligned at random.

Definition 2.9. *The odds ratio $o(YX)$ is a measure of how likely two aligned bases are to be the result of related sequences as opposed to unrelated sequences and it is defined as the ratio of the substitution probability $p(Y \cap X)$ over the expected probability by chance $e(Y \cap X)$.*

$$o(YX) = \frac{p(Y \cap X)}{e(Y \cap X)}$$

The new element introduced with the odds ratio is the expected frequency from random alignment $e(Y \cap Y)$. In the case of random alignment, the presence of base X in the reference and base Y in the read are independent from one another:

$$e(Y \cap X) = p(X) \times p(Y) \tag{2.7}$$

Applying Equations 2.1 and 2.7 transforms the definition of the odds ratio

to the following:

$$o(YX) = \frac{p(Y|X)}{p(Y)} \tag{2.8}$$

The frequency of Y in the mutated genome may not always be readily available. However, for very closely related genomes such as those of individuals of the same species, as is the case in genome resequencing, it can be assumed that the genome composition is the same. Indeed, it has been reported that the average frequency of substitutions among human individuals is $p(M) = 0.005$ [111].

Finally, all the prerequisite parameters have now been determined and the 4×4 scoring matrix for all outcomes (four matches and twelve mismatches) can be calculated:

Definition 2.10. *The score of a substitution is the logarithm of the odds ratio.*

$$s(YX) = log_2\big(o(YX)\big)$$

Alignment score

The score of an alignment between a read and the reference is the sum of the scores for the aligned bases at all the positions of the read, weighted by the confidence in the base call. The best alignment is the one with the highest score, instead of the one with the fewest mismatches.

Definition 2.11. *The score of an alignment between a read and the reference is the sum of the scores for the aligned bases at all the positions of the read, weighted by the error probability at each position [107].*

$$S = \sum_{i=1}^{\texttt{readLength}} \Big(s(Y_i X_i) \times \big(1 - P_Q(i)\big)\Big)$$

The score is specific to each combination of aligned bases, and an erroneous base call would lead to an erroneous score. Therefore, the higher the error probability, the lower the confidence in the score and the smaller its contribution

to the overall score.

Implementation

The substitution matrix described in the previous subsection was implemented as a C++ module. The incorporation of this scoring module into REAL was carried out by Dr. Tischler, who implemented the majority of *REAL*'s code. The module takes as parameters the transition bias $p(M_{ts}|M)$, the GC mutability bias B, the base composition of the genome $p(G \cup C)$ and the expected similarity level between the reference genome and the sequenced genome $(1 - p(M))$. With this information it initialises a 4×4 look-up table containing the substitution scores, as per Definition 2.10. During the alignment of reads, alignments are scored as per Definition 2.11 and the unique highest scoring alignment is reported.

2.3 Results

In order to test the usefulness of the proposed scoring model in the mapping of short NGS reads, the human chromosome 1, from assembly GRCh37 obtained from the NCBI GenBank database [113], was selected as the reference, simply because it is the largest of the human chromosomes. Five altered chromosome sequences were generated, one for each of the mutation rates 0.001, 0.005, 0.01, 0.05 and 0.1, using the mutation probabilities described in Section2.2.2 with the bias values reported for humans $B = 2$ and $p(M_{tv}|M) = 0.71$ [111]. From each of the four altered chromosomes, single–fold coverage was simulated, resulting in $6,257,754$ $36bp$–long reads, without any simulated errors, in order to test the scoring model under ideal conditions.

Additionally, from the altered chromosome at the mutation rate of 0.005, reads were generated for a range of error settings (see Section 2.2.3), in order to test the performance of the scoring model in more realistic situations:

1. Use of parameters $X_D = (\texttt{readLength} + 1)$ and $P_c = 0.000001$ focused on

the influence of the base call errors, with quickly deteriorating quality at the end of the reads.

2. Use of parameters $X_D = (10 \times \texttt{readLength})$ with a variatey of P_c values focused on the influence of the error rate that is not encoded in the base qualities, with high base call quality throughout the length of the reads.

3. Use of parameters $X_D = (\texttt{readLength} + 1)$ and $P_c = 0.01$ allowed both error types to work in conjunction.

4. Use of parameters $X_D = (\texttt{readLength} + 1)$ and $P_c = 0.05$ presented an extreme case of high error rate.

The reads were subsequently mapped to the original GRCh37 chromosome 1 sequence using the in-house algorithm REAL, modified with the scoring scheme described in Section 2.2.5. Each set of reads was aligned using three scoring models:

1. Use of parameters $B = 2$ and $p(M_{tv}|M) = 0.71$ created substitution scores based on the same substitution probabilities that were used to generate the mutations.

2. Use of parameters $B = B_{neut} = 0.7$ (Equation 2.2) and $p(M_{tv}|M) = 0.333$ created substitution scores based on the combinatoric substitution probabilities of a bias-free mutation model. These values were derived from retaining only the composition bias of the human genome $P(G \cup C) = 41\%$.

3. Finally, the complete by-pass of the scoring scheme reverted the algorithm to simply counting the number of mismatches.

The tests were repeated in an identical way under the same conditions with single–fold coverage sets of $3,128,861$ $72bp$–long reads, changing only *REAL*'s maximum number of allowed mismatches, from 5 to 10, in order to match the doubling of the read length. Despite the use of scores in the selection of the

best alignment, REAL uses the number of mismatches as a threshold, in order to reduce the number of candidate alignments that must be considered and speed up processing. The alignment seed length was kept to the default value of $32bp$ and the maximum number of mismatches in the seed was kept to the default value of 2 for both read lengths.

A second replicate of altered chromosomes was generated using the human GRCh37 chromosome 17, which has distinctly different length and average GC content, compared to chromosome 1, so as to verify that the observations were reproducible and not specific to chromosome 1 or to the mutations generated. Replicate 2 consisted of single–fold coverage sets of $2,160,970$ $36bp$–long reads and $1,080,484$ $72bp$–long reads, which were subjected to the same exact conditions as the chromosome 1 reads in Replicate 1.

2.3.1 Substitution Matrices

Prior to applying the proposed scoring scheme to actual alignments, a visual inspection of the scores generated was carried out. Table 2.1 presents the theoretical situation in which no biases would be present and the genome would be equally composed of all four bases. Tables 2.2, 2.3 and 2.4 present how the model changes under the influence of each of the biases individually, while the result of all the biases together is presented in Table 2.5. Finally, Table 2.6 presents the scores for different magnitudes of mutation rate, in the presence of all the biases.

The behaviour of the scores under the influence of each of the biases is the one that would be expected: A greater abundance of A and T led to greater scores for conservation of the rarer G and C, the higher frequency of transitions led to more lenient scores for transitions and more conservative scores for transversions, and the increased mutability of G and C led to their substitutions being more tolerated than the substitutions of A or T. Finally, different orders of magnitude for the mutation rate have a very strong effect on the scores.

substitution probabilities — $p(Y'\|X)$							
AA'	0.9900	AC'	0.0033	AG'	0.0033	AT'	0.0033
CA'	0.0033	CC'	0.9900	CG'	0.0033	CT'	0.0033
GA'	0.0033	GC'	0.0033	GG'	0.9900	GT'	0.0033
TA'	0.0033	TC'	0.0033	TG'	0.0033	TT'	0.9900
observed frequencies for related alignment — $p(Y' \cap X)$							
AA'	0.2475	AC'	0.0008	AG'	0.0008	AT'	0.0008
CA'	0.0008	CC'	0.2475	CG'	0.0008	CT'	0.0008
GA'	0.0008	GC'	0.0008	GG'	0.2475	GT'	0.0008
TA'	0.0008	TC'	0.0008	TG'	0.0008	TT'	0.2475
base composition — $p(X)$							
A	0.250	C	0.250	G	0.250	T	0.250
expected frequencies for random alignment — $e(Y' \cap X)$							
AA'	0.0625	AC'	0.0625	AG'	0.0625	AT'	0.0625
CA'	0.0625	CC'	0.0625	CG'	0.0625	CT'	0.0625
GA'	0.0625	GC'	0.0625	GG'	0.0625	GT'	0.0625
TA'	0.0625	TC'	0.0625	TG'	0.0625	TT'	0.0625
bit scores — $s(Y'X)$							
AA'	+1.99	AC'	−6.23	AG'	−6.23	AT'	−6.23
CA'	−6.23	CC'	+1.99	CG'	−6.23	CT'	−6.23
GA'	−6.23	GC'	−6.23	GG'	+1.99	GT'	−6.23
TA'	−6.23	TC'	−6.23	TG'	−6.23	TT'	+1.99

Table 2.1: Scoring scheme for mutation rate $p(M) = 0.01$ in the absence of biases: $p(G \cup C) = 0.5$, $B = B_{neut} = 1$, $p(M_{tv}|M) = 0.333$.

observed frequencies for related alignment — $p(Y' \cap X)$							
AA'	0.2921	AC'	0.0010	AG'	0.0010	AT'	0.0010
CA'	0.0007	CC'	0.2029	CG'	0.0007	CT'	0.0007
GA'	0.0007	GC'	0.0007	GG'	0.2029	GT'	0.0007
TA'	0.0010	TC'	0.0010	TG'	0.0010	TT'	0.2921
base composition — $p(X)$							
A	0.295	C	0.205	G	0.205	T	0.295
expected frequencies for random alignment — $e(Y' \cap X)$							
AA'	0.0870	AC'	0.0605	AG'	0.0605	AT'	0.0870
CA'	0.0605	CC'	0.0420	CG'	0.0420	CT'	0.0605
GA'	0.0605	GC'	0.0420	GG'	0.0420	GT'	0.0605
TA'	0.0870	TC'	0.0605	TG'	0.0605	TT'	0.0870
bit scores — $s(Y'X)$							
AA'	+1.75	AC'	−5.95	AG'	−5.95	AT'	−6.47
CA'	−6.46	CC'	+2.27	CG'	−5.94	CT'	−6.46
GA'	−6.46	GC'	−5.94	GG'	+2.27	GT'	−6.46
TA'	−6.47	TC'	−5.95	TG'	−5.95	TT'	+1.75

Table 2.2: Scoring scheme for mutation rate $p(M) = 0.01$ with a biased genome composition: $p(G \cup C) = 0.41$, $B = B_{neut} = 0.7$, $p(M_{tv}|M) = 0.333$.

observed frequencies for related alignment — $p(Y' \cap X)$							
AA'	0.2475	AC'	0.0004	AG'	0.0018	AT'	0.0004
CA'	0.0004	CC'	0.2475	CG'	0.0004	CT'	0.0018
GA'	0.0018	GC'	0.0004	GG'	0.2475	GT'	0.0004
TA'	0.0004	TC'	0.0018	TG'	0.0004	TT'	0.2475
base composition — $p(X)$							
A	0.250	C	0.250	G	0.250	T	0.250
expected frequencies for random alignment — $e(Y' \cap X)$							
AA'	0.0625	AC'	0.0625	AG'	0.0625	AT'	0.0625
CA'	0.0625	CC'	0.0625	CG'	0.0625	CT'	0.0625
GA'	0.0625	GC'	0.0625	GG'	0.0625	GT'	0.0625
TA'	0.0625	TC'	0.0625	TG'	0.0625	TT'	0.0625
bit scores — $s(Y'X)$							
AA'	$+1.99$	AC'	-7.43	AG'	-5.14	AT'	-7.43
CA'	-7.43	CC'	$+1.99$	CG'	-7.43	CT'	-5.14
GA'	-5.14	GC'	-7.43	GG'	$+1.99$	GT'	-7.43
TA'	-7.43	TC'	-5.14	TG'	-7.43	TT'	$+1.99$

Table 2.3: Scoring scheme for mutation rate $p(M) = 0.01$ in the presence of the transition bias: $p(G \cup C) = 0.5$, $B = B_{neut} = 1$, $p(M_{tv}|M) = 0.71$.

observed frequencies for related alignment — $p(Y' \cap X)$							
AA'	0.2483	AC'	0.0006	AG'	0.0006	AT'	0.0006
CA'	0.0011	CC'	0.2467	CG'	0.0011	CT'	0.0011
GA'	0.0011	GC'	0.0011	GG'	0.2467	GT'	0.0011
TA'	0.0006	TC'	0.0006	TG'	0.0006	TT'	0.2483
base composition — $p(X)$							
A	0.250	C	0.250	G	0.250	T	0.250
expected frequencies for random alignment — $e(Y' \cap X)$							
AA'	0.0625	AC'	0.0625	AG'	0.0625	AT'	0.0625
CA'	0.0625	CC'	0.0625	CG'	0.0625	CT'	0.0625
GA'	0.0625	GC'	0.0625	GG'	0.0625	GT'	0.0625
TA'	0.0625	TC'	0.0625	TG'	0.0625	TT'	0.0625
bit scores — $s(Y'X)$							
AA'	$+1.99$	AC'	-6.81	AG'	-6.81	AT'	-6.81
CA'	-5.81	CC'	$+1.98$	CG'	-5.81	CT'	-5.81
GA'	-5.81	GC'	-5.81	GG'	$+1.98$	GT'	-5.81
TA'	-6.81	TC'	-6.81	TG'	-6.81	TT'	$+1.99$

Table 2.4: Scoring scheme for mutation rate $p(M) = 0.01$ in the presence of the GC bias: $p(G \cup C) = 0.5$, $B = 2$, $p(M_{tv}|M) = 0.333$.

substitution probabilities — $p(Y'\|X)$							
AA'	0.9944	AC'	0.0008	AG'	0.0040	AT'	0.0008
CA'	0.0024	CC'	0.9837	CG'	0.0024	CT'	0.0115
GA'	0.0115	GC'	0.0024	GG'	0.9837	GT'	0.0024
TA'	0.0008	TC'	0.0040	TG'	0.0008	TT'	0.9944
observed frequencies for related alignment — $p(Y' \cap X)$							
AA'	0.2933	AC'	0.0002	AG'	0.0012	AT'	0.0002
CA'	0.0005	CC'	0.2017	CG'	0.0005	CT'	0.0024
GA'	0.0024	GC'	0.0005	GG'	0.2017	GT'	0.0005
TA'	0.0002	TC'	0.0012	TG'	0.0002	TT'	0.2933
base composition — $p(X)$							
A	0.295	C	0.205	G	0.205	T	0.295
expected frequencies for random alignment — $e(Y' \cap X)$							
AA'	0.0870	AC'	0.0605	AG'	0.0605	AT'	0.0870
CA'	0.0605	CC'	0.0420	CG'	0.0420	CT'	0.0605
GA'	0.0605	GC'	0.0420	GG'	0.0420	GT'	0.0605
TA'	0.0870	TC'	0.0605	TG'	0.0605	TT'	0.0870
bit scores — $s(Y'X)$							
AA'	+1.75	AC'	−7.97	AG'	−5.68	AT'	−8.49
CA'	−6.97	CC'	+2.26	CG'	−6.44	CT'	−4.68
GA'	−4.68	GC'	−6.44	GG'	+2.26	GT'	−6.97
TA'	−8.49	TC'	−5.68	TG'	−7.97	TT'	+1.75

Table 2.5: Scoring scheme for mutation rate $p(M) = 0.01$ in the presence of all the biases: $p(G \cup C) = 0.41$, $B = 2$, $p(M_{tv}|M) = 0.71$.

bit scores — $s(Y'X)$: $p(M) = 0.001$							
AA'	+1.76	AC'	−11.29	AG'	−9.00	AT'	−11.81
CA'	−10.29	CC'	+2.28	CG'	−9.76	CT'	−8.00
GA'	−8.00	GC'	−9.76	GG'	+2.28	GT'	−10.29
TA'	−11.81	TC'	−9.00	TG'	−11.29	TT'	+1.76
bit scores — $s(Y'X)$: $p(M) = 0.1$							
AA'	+1.68	AC'	−4.65	AG'	−2.35	AT'	−5.17
CA'	−3.65	CC'	+2.03	CG'	−3.12	CT'	−1.35
GA'	−1.35	GC'	−3.12	GG'	+2.03	GT'	−3.65
TA'	−5.17	TC'	−2.35	TG'	−4.65	TT'	+1.68

Table 2.6: Scoring scheme for mutation rates $p(M) = 0.001$ (top) and $p(M) = 0.1$ (bottom) in the presence of all the biases: $p(G \cup C) = 0.41$, $B = 2$, $p(M_{tv}|M) = 0.71$.

2.3.2 Effect of the score threshold

Before comparing the proposed scoring scheme with others, it was necessary to establish a criterion by which to decide if an alignment was sufficiently better than the next best one. When simply counting mismatches, the outcomes are quantized and such criterion is easy to implement; the unique best alignment is the one with the fewest mismatches and at least one less mismatch than the next best alignment. When, instead, a substitution matrix is used, the differences between alignment scores can be much more subtle.

As presented in Table 2.7, the lack of a threshold allows nearly all the reads to be aligned, but causes a high number of misalignments (in the order of 10% of aligned $36bp$–long reads). In order to determine the effect of setting a minimum threshold for the score difference between the two highest–scoring candidate alignments, values were tested across three orders of magnitude, from 0.05 to 5. This range was arbitrarily chosen and is centred around the value 0.5, which is in scale with the smallest difference between any two mismatch scores in a moderately conservative matrix (Table 2.5). This would be the difference between the alignment scores of two alignments with only one mismatch each and differing only in the type of the mismatch, a reasonable starting point in the search for a threshold.

The mere introduction of the threshold led to a drastic reduction of misalignments at all mutation rates, at the cost of some correct alignments. At the mutation rate of 0.005, between the threshold values of 0 and 0.5, $247,585$ fewer $36bd$–long alignments are obtained, of which the vast majority ($183,917$) would have been misalignments. Exploration of threshold values one order of magnitude below (0.05) and above (5) the starting point showed that further tuning of the threshold has comparatively little effect on the number of misalignments that are prevented. At the same mutation rate of 0.005, between the threshold values of 0.05 and 0.5, 901 fewer alignments are obtained with the higher threshold, of which more than half (553) would have been misalignments, a beneficial trade–off. However, between the threshold values of 0.5

Mutation Rate	*Score difference threshold* 0	0.05	0.125	0.5	2.5	5
36*bp*–long error–free reads from Chromosome 17 (2, 160, 970)						
						Total number of aligned reads
0.001	2160955	1912021	1912021	1911841	1910936	1910750
0.005	2159893	1913209	1913209	1912308	1907694	1906151
0.01	2153044	1909690	1909686	1907856	1898484	1894883
0.05	1733719	1544590	1542786	1535187	1500192	1445817
0.1	871268	772323	772316	764802	733423	696536
						Number of misaligned reads
0.001	186270	1448	1448	1328	915	889
0.005	190579	7205	7205	6662	4452	4166
0.01	195693	14492	14488	13320	8627	7831
0.05	207908	63607	62165	56366	34256	15286
0.1	152057	73317	73310	67050	42675	23102
72*bp*–long error–free reads from Chromosome 17 (1, 080, 484)						
						Total number of aligned reads
0.001	1080475	1038915	1038915	1038887	1038791	1038808
0.005	1080475	1038915	1038915	1038887	1038791	1038806
0.01	1076286	1034757	1034756	1034575	1033513	1033159
0.05	858379	824929	824624	823331	817322	806429
0.1	410225	393828	393813	392403	386298	376084
						Number of misaligned reads
0.001	25186	115	115	93	61	63
0.005	25186	115	115	93	61	64
0.01	26406	1332	1331	1229	825	727
0.05	34328	13613	13364	12380	8819	5921
0.1	26745	16287	16276	15162	10828	7140

Table 2.7: Influence of setting a threshold to the score difference between the best and second best alignments of a read.

and 5, 6, 157 fewer alignments are obtained with the higher threshold, of which less than half (2, 496) would have been misalignments, an expensive trade–off. Thus, the threshold value of 0.5 appears to be on the tipping point, between gaining more alignments and suffering increased misalignments. The same tipping point is observed with the longer 72*bp* reads, although the overall fraction of misalignments is drastically smaller.

The reads used for the determination of a threshold were obtained from the human assembly GRCh37 chromosome 17 across a range of mutation rates. They were also free of all simulated errors, as errors at this stage would only serve to confound the results.

2.3.3 Effect of the substitution model

In order to test the influence of the substitution parameters used in the creation of the scoring matrix, the sets of reads used in this subsection are all completely free of simulated errors and all the base mismatches in the alignments are the result of simulated substitution events. The performance of scores based on the biased substitutions model was compared to the performance of scores based on unbiased substitutions and the performance of counting the mismatches.

The results presented in Tables 2.8 and 2.9 demonstrate that the use of a substitution matrix (both the biased and the unbiased model) is able to map more reads than the simple count of mismatches. Out of the two matrix-based models, the biased one maps more reads than the unbiased model. However, a portion of these additionally mapped reads are incorrectly aligned. In mutation rates up to 0.01, these misalignments represent less than half the alignments gained by use of the biased model, instead of the unbiased or mismatch model. This is the same for both 36*bp*–long read replicate sets (Table 2.8). With the 72*bp*–long read sets (Table 2.9), the advantage of the biased model extends further, to the mutation rate of 0.05 in both replicates. Beyond the mutation rate of 0.01 for 36*bp*–long reads and 0.05 for 72*bp*–long reads, the additional reads mapped by the biased model are mostly misaligned, limiting the scoring model to closely related sequences.

2.3.4 Effect of errors

In order to investigate the influence of the base call errors (P_Q) versus that of errors of other origins (P_c), the same mutation rate (0.005) was used for all the read sets generated. The results are presented in Tables 2.10 and 2.11 (36*bp* and 72*bp*–long reads respectively). As expected, the increase in error rate causes fewer reads to be aligned altogether and a larger proportion of them to be misaligned. In all cases, the biased model maps the most reads out of the three examined models, but the mismatch model has the fewest misalignments

Mutation Rate	*Biased model*	*Unbiased model*	*Mismatch model*
Replicate 1 — $6,257,754$ $36bp$–long reads from Chromosome 1			
Total number of aligned reads			
0.001	$5,628,920$	$-1,139$	$-3,003$
0.005	$5,630,764$	$-5,594$	$-14,789$
0.01	$5,617,946$	$-10,787$	$-28,502$
0.05	$4,518,270$	$-29,509$	$-89,909$
0.1	$2,245,189$	$-53,325$	$-77,020$
Number of misaligned reads			
0.001	$3,547$	-406	$-1,218$
0.005	$17,669$	$-2,184$	$-6,386$
0.01	$35,181$	$-4,227$	$-12,645$
0.05	$147,890$	$-14,411$	$-52,202$
0.1	$171,397$	$-37,978$	$-55,486$
Replicate 2 — $2,160,970$ $36bp$–long reads from Chromosome 17			
Total number of aligned reads			
0.001	$1,911,859$	-436	$-1,162$
0.005	$1,912,324$	$-2,293$	$-5,790$
0.01	$1,907,923$	$-4,201$	$-11,093$
0.05	$1,535,513$	$-11,595$	$-34,611$
0.1	$765,327$	$-21,200$	$-30,051$
Number of misaligned reads			
0.001	$1,389$	-161	-488
0.005	$6,699$	-901	$-2,506$
0.01	$13,405$	$-1,601$	$-4,981$
0.05	$56,527$	$-5,510$	$-20,178$
0.1	$66,789$	$-14,977$	$-21,676$

Table 2.8: Influence of the mutation biases on $36bp$–long reads. The unbiased and mismatch model measurements are presented relatively to the biased model.

Mutation Rate	*Biased model*	*Unbiased model*	*Mismatch model*
Replicate 1 — 3,128,861 72*bp*–long reads from Chromosome 1			
Total number of aligned reads			
0.001	3,007,178	−142	−515
0.005	3,007,178	−142	−515
0.01	2,996,706	−1,333	−5,101
0.05	2,384,894	−3,651	−19,176
0.1	1,139,666	−10,267	−17,335
Number of misaligned reads			
0.001	577	−44	−215
0.005	577	−44	−215
0.01	5,777	−332	−2,204
0.05	38,174	−859	−11,094
0.1	44,164	−6,479	−11,665
Replicate 2 — 1,080,484 72*bp*–long reads from Chromosome 17			
Total number of aligned reads			
0.001	1,038,898	−30	−109
0.005	1,038,898	−30	−109
0.01	1,034,668	−321	−1,317
0.05	823,111	−913	−5,708
0.1	391,934	−3,161	−5,566
Number of misaligned reads			
0.001	93	−2	−32
0.005	93	−2	−32
0.01	1,258	−44	−501
0.05	12,088	−83	−3,119
0.1	15,210	−1,879	−3,601

Table 2.9: Influence of the mutation biases on 72*bp*–long reads. The unbiased and mismatch model measurements are presented relatively to the biased model.

of the three.

Case (c), where $P_c = 0.000001$, demonstrates the influence of base call errors under a negligible rate of errors not encoded in the qualities. The base quality is set to decrease drastically by the end of every read, guaranteeing a high error rate. All three models show only a small increase in misalignments, compared to their respective results with the error–free dataset, indicating a good ability to take the quality into account. However, both the unbiased and the mismatch models align fewer reads than the biased model, the majority of which would have been correct alignments. For example, when considering the 36*bp*–long reads in Replicate 1 (Table 2.10), the unbiased model had $6,481$ fewer alignments than the biased, but only $2,802$ fewer misalignments, representing a difference of $3,679$ correct alignments that the biased model did detect. A similar situation occurs with the 72*bp*–long reads (Table 2.11) as well as with Replicate 2 (Tables 2.10 and 2.11). Similar behaviour is also observed for the mismatch model. Thus, in the presence of only quality–encoded errors the biased model is a better choice.

Cases (a) and (b), where $X_D = 360$ or $X_D = 720$ (for 36*bp* and 72*bp*–long reads respectively), put the de-synchronisation length far beyond the end of the reads, ensuring high quality throughout their length and focusing on the influence of the errors that are not accounted for in the base quality. All models show a decrease in overall alignments, which is most pronounced in the mismatch model (reaching $-70,927$ in case (b) of the 36*bp*–long reads of Replicate 1 (Table 2.10), considerably more than the $-42,985$ of the biased model. However, the simultaneous increase in misalignments is most pronounced in the biased model ($+40,323$ for the same case), considerably more than the mismatch model's $+17,938$. When comparing the mismatch model to the biased model directly, the differences are indecisive when the error rate is low (case (a)), but favour the mismatch model when the error rate is higher (case (b)). For example, in the case of the 36*bp*–long reads of Replicate 1, in case (a), the mismatch model aligns $18,129$ fewer reads than the biased model, half of which

are misalignments. But in case (b), the misalignments account for more than half of the alignments lost in the mismatch model. The situation is similar when comparing the unbiased model to the biased one. Similar results are shown for both replicates at both read lengths. This finding is important and anticipated, as it validates that the scoring schemes work properly. The errors introduced by P_c are random (unbiased). When one considers that the mutation rate simulated is 0.005, it becomes evident that in case (b) the error rate (0.01) is considerably higher than the mutation rate and should have a more dominant influence. Thus, unbiased models are expected to fare better, which indeed they both do. In case (a), the error rate (0.001) is lower than the mutation rate, yet it holds a strong influence. Although the biased model aligns more reads than the other two models, half of the gained alignments are erroneous.

Finally, cases (d) and (e) present a combination of high base–call errors and high external constant errors. Case (d) results are very similar to those of case (b). The high constant error rate overwhelms the mutation rate, and favour the unbiased and mismatch models. The base–call errors are effectively compensated for by all models. Case (e) is a very extreme case with a very high constant error rate, further magnifying the results of case (d).

2.4 Discussion and Conclusion

In this chapter, a substitution matrix based on log odds ratios was derived from biases in the observed substitution frequencies in humans and was subsequently compared to the widely used practice of simply counting mismatches in the read alignments. A range of mutation rates were simulated, as well as a range of error rates. Under all the conditions that are relevant to human genome re-sequencing, the proposed biased model performed better than the mismatch model. Only in the presence of high error rates that are not reflected by the base qualities is the mismatch model preferable.

The reads used in the comparison were simulated using the same realistic

Error Rate	*Biased model*	*Unbiased model*		*Mismatch model*	
Replicate 1 — 6,257,754 36*bp*–long reads from Chromosome 1					
Total number of aligned reads					
none	5,630,764	5,625,170	(−5,594)	5,615,975	(−14,789)
(a)	−59	−1,226	(−6,761)	−3,399	(−18,129)
(b)	−42,985	−52,726	(−15,335)	−70,927	(−42,731)
(c)	+752	−135	(−6,481)	−492	(−16,033)
(d)	−44,845	−55.369	(−16,118)	−73,147	(−43,091)
(e)	−1,344,860	−1,371,832	(−32,566)	−1,426,187	(−96,116)
Number of misaligned reads					
none	17,669	15,485	(−2,184)	11,283	(−6,386)
(a)	+4,658	+3,685	(−3,157)	+2,030	(−9,014)
(b)	+40,323	+31,886	(−10,621)	+17,938	(−28,771)
(c)	+2,025	+1,407	(−2,802)	+1,098	(−7,313)
(d)	+41,342	+32,274	(−11,252)	+18,456	(−29,272)
(e)	+139,571	+114,631	(−27,124)	+68,872	(−77,085)
Replicate 2 — 2,160,970 36*bp*–long reads from Chromosome 17					
Total number of aligned reads					
none	1,912,324	1,910,031	(−2,293)	1,906,534	(−5,790)
(a)	−171	−620	(−2,742)	−1,419	(−7,038)
(b)	−14,538	−18,3644	(−6,119)	−15,786	(−16,486)
(c)	+161	−166	(−2,620)	−390	(−6,341)
(d)	−15,134	−19,554	(−6,713)	−26.360	(−17,016)
(e)	−453,687	−464,541	(−13,147)	−485,126	(−37,229)
Number of misaligned reads					
none	6,699	5,798	(−901)	4,193	(−2,506)
(a)	+1,666	+1,337	(−1,230)	+695	(−3,477)
(b)	+15,144	+11,836	(−4,209)	+6,606	(−11,044)
(c)	+730	+474	(−1,157)	+309	(−2,927)
(d)	+15,569	+11,798	(−4,672)	+6,663	(−11,412)
(e)	+53,547	+43,512	(−10,936)	+26,480	(−29,573)

Table 2.10: Differences in total number of mapped reads and in number of misaligned reads caused by the introduction of errors on 36*bp*–long simulated reads, with a mutation rate of 0.005. The absolute read counts are given for the error–free case. Error case results are presented relative to the respective error–free case for each of the three models. The numbers in parentheses indicate how the unbiased and mismatch models fared relative to the biased model.
none: completely error–free reads,
(a): $X_D = 360$, $P_c = 0.001$,
(b): $X_D = 360$, $P_c = 0.01$,
(c): $X_D = 37$, $P_c = 0.000001$,
(d): $X_D = 37$, $P_c = 0.01$,
(e): $X_D = 37$, $P_c = 0.05$.

Error Rate	*Biased model*	*Unbiased model*		*Mismatch model*	
Replicate 1 — 3,128,861 72*bp*–long reads from Chromosome 1					
Total number of aligned reads					
none	3,007,178	3,007,036	(−142)	3,006,663	(−515)
(a)	−1,743	−2,477	(−876)	−4,295	(−3,067)
(b)	−32,418	−34,266	(−1,990)	−39,372	(−7,469)
(c)	−898	−1,643	(−887)	−3,245	(−2,862)
(d)	−32,629	−34,575	(−2,088)	39,613	(−7,499)
(e)	−744,826	−748,972	(−4,288)	−762,693	(−18,382)
Number of misaligned reads					
none	577	533	(−44)	362	(−215)
(a)	+3,107	+2,815	(−336)	+1770	(−1,552)
(b)	+9,264	+8,019	(−1,289)	+4,452	(−5,027)
(c)	+2,632	+2,322	(−354)	+1,498	(−1,349)
(d)	+9,301	+8,006	(−1,339)	+4,516	(−5,000)
(e)	+35,307	+31,881	(−3,470)	+21,043	(−14,479)
Replicate 2 — 1,080,484 72*bp*–long reads from Chromosome 17					
Total number of aligned reads					
none	1,038,898	1,038,868	(−30)	1,038,789	(−109)
(a)	−997	−1,154	(−187)	−1,632	(−744)
(b)	−12,183	−12,568	(−415)	−13,873	(−1,799)
(c)	−646	−835	(−219)	−1,253	(−716)
(d)	−12,135	−12,581	(−476)	−13,882	(−1,856)
(e)	−258,348	−259,419	(−1,101)	−263,423	(−5,184)
Number of misaligned reads					
none	93	91	(−2)	61	(−32)
(a)	+583	+550	(−35)	+306	(−309)
(b)	+1,934	+1,892	(−44)	+946	(−1,020)
(c)	+547	+547	(−2)	+300	(−279)
(d)	+2,016	+1,935	(−83)	+988	(−1,060)
(e)	+10,540	+8,663	(−1,879)	+6,724	(−3,848)

Table 2.11: Differences in total number of mapped reads and in number of misaligned reads caused by the introduction of errors on 72*bp*–long simulated reads, with a mutation rate of 0.005. The absolute read counts are given for the error–free case. Error case results are presented relative to the respective error–free case for each of the three models. The numbers in parentheses indicate how the unbiased and mismatch models fared relative to the biased model.
none: completely error–free reads,
(a): $X_D = 720$, $P_c = 0.001$,
(b): $X_D = 720$, $P_c = 0.01$,
(c): $X_D = 73$, $P_c = 0.000001$,
(d): $X_D = 73$, $P_c = 0.01$,
(e): $X_D = 73$, $P_c = 0.05$.

biases used in the creation of the scoring matrix. Two read lengths were used; the length of $36bp$ was the standard Illumina output length in 2008, whereas the length of $72bp$ was at the time under development and has since been surpassed. Technology and chemistry improvements in the NGS platforms are constantly pushing the limits towards greater lengths, whilst the still emerging third generation of sequencing technologies promises much longer reads [114] which likely will not require mapping to a reference.

The reads were simulated in such a way that they would provide exactly $1\times$ coverage of the reference sequence. Thus, each and every locus on the reference corresponded to exactly one read, and any observed differences in the numbers of aligned and misaligned reads would explicitly indicate the number of loci for which information could be obtained by the alignments. This made the effect of substitutions and errors easier to quantify.

The mutation rates tested were limited to a range between 0.1 and 0.001, knowing that the rate among humans is in the order of 0.005. The scoring model makes the assumption that the overall base composition of the genome does not change significantly, which restricts its application to alignments within the same or very closely related species. As expected, the ability of the algorithm to map reads, under any of the scoring schemes, decreased when more substitutions were present.

Two error types were considered: Errors that are encoded in the base qualities and errors that are not. The former can be caused by progressive loss of synchronisation among the cloned fragments, as sequencing cycles progress, leading to increased errors by the end of each read. They can also be caused at any position as the result of a replication error that propagated among the clones. Errors that are not encoded in the base qualities might be caused by other manipulations of the sequence fragments, prior to amplification. Errors encoded by the base qualities can be effectively compensated for, whereas errors not encoded by the qualities have a significant impact. When the rate of random unbiased errors is higher than the mutation rate, more mismatches are due to

these errors than due to the mutations and, thus, the substitution frequencies modelled by the biased scoring scheme no longer reflect the actual frequencies of substitution. Instead, the unbiased error frequencies are better modelled by the mismatch or unbiased schemes.

In any case, high error rates that are not encoded in the base qualities (P_c), would be indicative of very poor data quality caused by poor sample handling and preparation. In reality, one would not expect such high P_c rates, but they were included in the trials because they effectively alter the substitution frequencies in a way similar to having used an unbiased mutation rate for the read generation. The fall in performance when the biased model is used on high error reads, together with the fall in performance when the unbiased model is used on biased reads, highlights the importance of matching the model to the actual substitution frequencies, even with short NGS reads.

Finally, the mismatch model offers a simple way to distinguish the best alignment, but it cannot distinguish between alignments with the same number of mismatches. The use of a scoring matrix can break such ties. However, this may not always lead to a meaningful result, as demonstrated by the high number of misalignments in Table 2.7. The use of a threshold for the difference between the alignment scores is a simplistic approach that proves to be effective.

Although the scoring matrices presented in this chapter were built based on the human values for the biases and all the simulation data was from the human genome, there is no limitation to do so. The matrices are calculated at runtime, based on the parameters supplied. By inputting the appropriate bias values, the scoring scheme can be effortlessly adapted to accommodate the conditions found in other organisms.

Similar works

The use of a substitution matrix has been proposed before [52, 107].

The former work, [52], presents a dynamic programming theoretical algorithm for gapped alignment of two weighted sequences to one another. There

is one single score for matches of any kind, one for mismatches of any kind and one for gaps. In that sense, the matrices are in fact not substitution matrices, but rather probability matrices describing the weighted sequences. The score for two aligned positions is the sum of the probabilities of the up to 25 possible combinations (4 bases and 1 gap possibility per position), each factored by one of the three scores (match,mismatch,gap). Sequences with base qualities are converted to weighted sequences by splitting the error probability evenly among to the non–match possibilities (mismatches or gap). The algorithm could easily be converted to use a substitution matrix instead of its base–insensitive scores, but the validity of its scoring formulation has been questioned [107]. Indeed, the is intuitive but arbitrary, without theoretical justification and the simplistic weighting mechanism would not handle correctly the weight of scores that reflected sequence similarity. The algorithm also misses the opportunity to bypass base qualities altogether in order to work directly with the raw sequencing signal, in which case it would have presented an improvement over Slider [21], since the latter deals with raw base signals but does not allow for scores.

The work by Frith *et al* [107], combines a true substitution matrix with quality scores and was tested on $36bp$ and $51bp$–long simulated gap-less reads from the human chromosome 1. The tests carried out are largely similar to the ones carried out in this chapter and both conclude that the use of substitution matrices reflecting sequence evolution is preferable even in the case of short NGS reads. The two works differ in the substitution matrices used, as the work presented here also takes into account the increased mutability of G and C and the base composition bias of the genome, in addition to the transition bias and mutation rate considered by both works. Another difference is the scope of the matrices. The present work focuses on genome resequencing and assumes near identical genome composition between the reference and the sequenced genomes, whereas the work by Frith *et al* aims at cross–species alignments. Furthermore, the influence of errors that are not reflected by the base qualities is measured, in addition to the influence of base call errors measured by both

works. Finally, the systematic process for the creation of simulated reads in the present work allows for the quantification of the impact of mutations and errors on the number of genomic loci that can be detected by the read aligner. On the contrary, the number of reads sampled in the previous work was too small to achieve complete coverage of the chromosome.

This chapter also addressed the problem of read simulation, based on specific substitution probability requirements. Other simulation scripts are available [42,67], but they lack the finer control of the substitution biases and the genome coverage. The calculation of the substitution probabilities was required for the alignment scores, and the creation of a custom read–simulation script was a small overhead, compared to the benefit of complete control and confidence over the parameters and method used.

Perspectives

The results presented in this chapter disregard the influence of gaps in the alignments; all the simulated reads are free of insertions and deletions. In real practice, insertions and deletions (indels) are not uncommon and post-transcriptional processing of RNAs introduces further alignment gaps where exons were spliced together. Although these are considered to various extents by other alignment algorithms, *REAL* in its initial form did not. The algorithm is still being extended but, although some provisions have been made for gaps [65], the feature is not yet complete.

In keeping with the idea of tailoring a substitution matrix to the observed substitution frequencies, it would be interesting to explore whether tailoring the gap lengths for exome sequencing could have similar benefits, once *REAL*'s gapped alignment feature has been completed. Gaps in exome sequencing tend to have length of the type $3n\,,\ n \in \mathbb{N}$, as a result of the genomic code. Gap lengths of this type lead to amino acid insertions and deletions, but leave the rest of the amino acid sequence unchanged. On the contrary, lengths of n or $2n$ are frame–shifting and alter the entire amino acid sequence after that point,

thus they are more likely to be damaging and be negatively selected.

Another possible extension would be to consider alignments of more distant sequences. The substitution probabilities calculated in the present work were based on the assumption of very closely related sequences, where the genome composition would be the same. However, the mapping algorithm would easily work with any other substitution matrix and the read simulation script could be modified to account for less similar sequences.

Other extensions of the algorithm involve the use of a circular reference sequence [115] and the use of multiple reference sequences [116]. The former addresses the alignment of reads to bacterial genomes and plasmids, while the latter aims at mapping reads to one of a number of closely similar reference sequences (for example different individuals of the same species) as a means of incorporating known sequence variation into the alignment stage and reduce misalignments. Neither makes use of the proposed scoring scheme in its initial release, but the modification to incorporate a substitution matrix would be simple.

Finally, this work only considered the effect of different substitution models on the alignment at the level of populations of reads. While this is enough to prove whether a model is better or not, as was this chapter's mission, interesting insights might be gained by exploring the correlation between the models and identifying which reads were affected by the use of the different models. This would answer questions regarding the properties of the affected reads in comparison to the unaffected ones, such number of substitutions, number of errors and presence of repeats and explain better why one model works better than another.

Conclusion

In spite of most read mapping algorithms using a simple match/mismatch model to judge alignments and some of them also consider the base qualities, this chapter and similar work provides sufficient evidence in favour of modelling the ob-

served substitution frequencies, even at the low mutation rates observed among individuals of the same species. Furthermore, this benefit is more pronounced in short reads, which are the most difficult to map.

Although it is likely that NGS technologies will be eventually replaced by the newer generation of sequencing platforms which will be able to sequence much longer reads, this transition has not yet begun, therefore any improvements to the analysis tools tied to the NGS pipelines are still relevant.

Chapter 3

Application of REAL to the investigation of the relationship between expression and the genomic base composition

3.1 Background

3.1.1 What are isochores

Isochores were discovered as strata obtained by density gradient centrifugation of fragmented vertebrate genomes [117]. They indicate that base composition is not uniform along the entirety of a vertebrate genome. Instead, vertebrate genomes are a mosaic of compositional domains, called isochores, each of which represents a region of locally homogeneous base composition [118, 119] with

distinct boundaries [120]. Isochore length can vary from $0.2Mb$ up to several Mb [121,122] and the local base composition ranges from 30% to 60% guanine-cytosine (GC). The isochores have been observed to cluster into five families, based on their base composition, and labelled L1, L2, H1, H2 and H3, from the lowest average GC% (lighter – L) to the highest GC% (heavier – H) [118]. This grouping in five families can be clearly seen in the figures of this chapter (Figs. 3.1, 3.2, 3.4, 3.5).

These compositional domains in the vertebrate genomes have been found to coincide with a range of structural and functional properties which also display mosaic distribution in the genome [123]. Isochores rich in GC are typically associated with high gene density [118, 124–126], high level of expression [126, 127], higher density of short repetitive elements (SINES) and low density of long repetitive element (LINES) [118,126], early replication [118,128], higher level of recombination and higher rate of mutations [129]. It has also been shown that the compositional compartmentalisation of the genome into isochores is reflected by the chromosomal bands displayed after Giemsa staining [118, 130, 131].

Additionally to the bias in the overall gene distribution in the different isochore families, a bias in the function of the genes and control sequences associated with different GC content has been observed. More specifically, programmed changes in both the expression level and the replication timing which are observed during development and cell differentiation are concentrated on genes found in GC-poor isochores [132]. In addition, genes expressed early in development tend to have AT-ending codons [133], which are usually located in GC poor isochores [134]. It has also been observed that broadly expressed "housekeeping" genes have a distribution of base composition that is skewed towards high GC levels, whereas tissue-specific genes are on average slightly GC-poorer [135]. Furthermore, the sequence context of the translation initiation codon differs based on the GC content of the isochores, possibly influencing the efficiency of transcription [136], and the transcription promoters found in GC-poor isochores differ from those found in GC-rich regions [137] and tend to

be associated with genes of different functional groups [138]. Similarly, genes of different functional classes present different *GC* content [139]. Finally, isochores can also affect gene regulation at the level of chromatin structure, such as nucleosome positioning [140]. These studies suggest that the compositional compartmentalisation of the genome has a functional role in the control of chromatin structure and gene regulation [140–143].

3.1.2 Evolution of isochores

Although the mechanisms that created and maintain the isochores and connect them with the various correlated features are beyond the scope of the present work, a brief overview would help place this work in the context of the ongoing debate.

Two main competing views exist with regards to isochore evolution. The first view, supports that isochores were formed under selective pressure [118], whereas the other supports that isochores are a by-product of biased mutational processes [144,145]. Upon release of the human genome's sequence, a third view appeared, claiming that isochores didn't exist at all, but were instead an artefact of the methodologies used, the vague definition of what isochores were and the subjectivity of the researchers in defining isochore borders [146, 147], all of which prevented independent confirmation of isochore sizes and locations using the newly known detailed genome sequence. As of the current time, the controversy still holds, and all sides regularly release updates to refute criticism against them [134,142,143,148]. However, given that compositional partitioning of vertebrate genomes into isochores correlates with a large number of features that are also non-uniformly distributed along the genome, including the visual bands obtained by Giemsa staining of chromosomes, it is safe to conclude that isochores do, in fact, exist in some form [142]. Indeed, this compositional compartmentalization has recently been extended to all eukaryotes [149].

Much of the controversy between the competing theories for isochore evolution can be attributed to the fact that the compositional partitioning of ver-

tebrate genomes correlates with a large number of other features, allowing abundant space for different interpretations, which are not necessarily mutually exclusive. The selectionist model proposes that the higher GC content in transcriptionally active regions enhances the thermodynamic stability of DNA, RNA and proteins. Thus the increase in GC was the result of evolutionary pressure to counterbalance the increase in body temperature [118,141] and has been maintained across taxa due to this pressure. However, this theory offers no concrete molecular mechanism that would create the increase in GC, in light of the inherently increased tendency of GC to mutate towards AT, and specifically the tendency of methylated C to convert into T. Despite lacking a concrete mechanism and despite criticism that the increase in GC pre-dates homeothermy and does not correlate with temperature [150, 151], it has been confirmed across a broad spectrum of organisms that hot environment dwellers have GC-enriched genomes, compared to organisms who prefer colder environments [141]. The opposing theories suggest that the creation and maintenance of GC-rich regions is a mere by-product of either local variation in mutation rates [144] or biased gene conversion [142, 145]. The theory that involves differing mutation rates has been conclusively refuted [152]. The remaining two theories (thermodynamic stability versus biased gene conversion), though apparently conflicting, are not mutually exclusive.

Indeed, Bernardi [143] admits that biased gene conversion may be one of the molecular mechanisms that created and maintains the increase in GC. Another point of disagreement is the plausibility of selection acting on the relationship between composition and gene expression, because the change in GC content affects both coding and non-coding regions [142]. Although gene regulation may not have been the selective factor in isochore evolution, there is a clear connection between composition, chromatin compaction and nucleosome positioning [123, 140, 141], which in turn are connected with gene expression at a scale much larger than local regulatory elements and gene sequences. Indeed, the original suggestion was exactly the reverse relationship; namely that higher

GC content is advantageous to highly expressed gene–dense regions, because the chromatin there is unpacked and thus more vulnerable to strand separation [118].

Currently, both theories lack sufficient data to conclusively refute one another and the controversy continues. Since, however, they are not as mutually exclusive as they claim to be, it may never be possible to completely refute one or the other. Instead, the truth may lie in a fusion of the two theories, and at least one of the involved parties has moved in that direction [143].

3.1.3 The present work

The correlation between the base composition and the expression level of human genes has been investigated extensively, using data from various techniques based on hybridisation (micro-arrays) or sequencing (ESTs, SAGE, MPSS) [123, 126, 127, 135, 153–160], and similar results have been confirmed for the mouse and other mammals [161, 162].

The techniques used in these studies have shortcomings that influence the measured correlations, such as a limited number of studied genes. In that aspect, NGS is an attractive approach due its constantly decreasing cost and its potential for generation of very large amounts of data. RNA sequencing using NGS techniques (RNA-seq) can yield high read coverage and does not require the prior knowledge of specific gene locations or marker sequences, enabling the potential detection of previously unknown or alternatively spliced transcripts. Furthermore, read coverage does not present saturation effects, enabling the detection of rare transcripts through the increase of sequencing depth. Up to the point of the present work, no studies had established the correlation between expression and composition using data from NGS techniques. Additionally, despite the extensive research on the implication of base composition in gene regulation, no transcriptome map had been drawn focusing on the isochores until this work. Given the apparent importance and implication of isochores in transcription regulation during cell differentiation and organism development,

a transcriptome map focusing on isochores instead of genes would offer a better overview of this relationship.

Thus, in this chapter, *REAL* (see 1.2.3) was used to draw an isochoric transcriptome map and investigate the relationship between the base composition of isochores and the expression level of genes in three distinct tissues of the adult mouse (brain, liver, muscle), as well as two developmental stages of the mouse brain (embryonic day 18, post-natal day 7), using publicly available RNA-seq data. The results confirm the previous reports of a positive correlation between *GC* content and expression and provide a more integrated view on the effect of base composition on transcription.

3.2 Materials and Methods

3.2.1 Data and alignment

To produce the transcriptome map of the isochores, publicly available RNA-seq data from three distinct adult mouse tissues (brain, liver, muscle [163]) and two developmental stages of the mouse brain (embryonic day 18, post-natal day 7 [164]) was used. The data was produced on Solexa platforms and is thus suitable for use with *REAL*. Prior to alignment, the short single-end reads were truncated to a length of 25 bases, in order to remove low-quality base-calls near the end.

Gap-less alignment with a maximum of two base mismatches was performed using *REAL* to map the reads onto the mouse reference genome, UCSC release mm9 [165] (Table 3.1). Reads that were mapped equally well to multiple locations, were discarded. The choice of *REAL* for this task was justified by the inefficiency of other contemporary popular fast aligners (Bowtie [48], SOAP2 [49]) at dealing with the very short read lengths of this dataset. The version of *REAL* used did not incorporate the scoring scheme described in the previous chapter, as these works took place in parallel and the scheme was not finalized at the time.

Although gap-less alignment of RNA-seq reads prevents the alignment of reads that span splice sites, given the very short length of the reads, the fraction of them that span splice sites is expected to be small (in the order of 1 splice site for every 1000 reads [163]). Furthermore, gapped alignment of such short single-end reads would likely result in a steep rise of ambiguously mapped reads. As the aim of this study is not the precise quantification of individual RNAs and alternative transcripts, but, instead, the overall picture of expression on a broad genomic scale, this sacrifice in sensitivity will cause lower perceived expression levels across the whole genome, therefore it is likely to have little impact on the relative expression levels that are of interest here.

3.2.2 Expression level of isochores

To measure the expression levels of mouse isochores, the reads aligned by *REAL* were cross–referenced against the positions of the isochores ([119]). Isochore expression was measured individually for each isochore, so as to create an isochore–centric expression map along the genome. The isochores were then binned together in five families according to their base composition ([119]), so as to obtain the average expression by isochore family.

Multiple factors can affect the number of reads aligned to an isochore. Firstly, fluctuations in the sequencing depth affect the overall number of reads generated and render impossible the direct comparison of read counts from different sequencing runs. Also, in a theoretical situation where reads were evenly distributed along the genome, larger isochores would have more reads by mere virtue of their greater size, without this implying anything about their relative intensity of expression. Neither of these two influences is relevant in the study of the relationship between base composition and expression level. In order to eliminate their influence, the absolute read counts per isochore were normalised for the total read count for each tissue and for the length of the respective isochores, producing thus a relative measure of expression level that is comparable across tissues and isochores. However, as the values produced are minuscule and

unintuitive, a scaling factor (10^6) was applied. Equation 3.1 summarises this process (E_{iL}: expression level of isochore i normalised for isochore length and sequencing depth, R_i: read count for isochore i, R_t: total read count for the tissue or developmental stage, L_i: length of isochore i, f: scaling factor). Thus, E_{iL} roughly represents the number of reads that would have been aligned, if the isochores had all the same size and the sequencing depth was exactly the same for all tissues.

$$E_{iL} = \frac{R_i}{R_t \times L_i} \times f \tag{3.1}$$

The number of aligned reads in an isochore is also directly influenced by the gene density of the isochore. As was mentioned in Section 3.1, *GC*-richer isochores tend to have higher gene densities (Fig. 3.2). Equation 3.2 removes this influence on the expression, in order to produce a clearer picture of the direct relationship between the base composition and the expression, if one exists (E_{iG}: expression level of isochore i normalised for both the length and the gene content of the isochore, G_i: number of genes in isochore i). The number of genes is used in the normalisation, instead of the gene density, because the expression E_{iL} is already normalised for isochore size.

$$E_{iG} = \frac{R_i}{R_t \times L_i \times G_i} \times f = \frac{E_{iL}}{G_i} \tag{3.2}$$

Similarly, in order to highlight the effect of gene density on the expression level of an isochore, the latter must be normalised for the *GC* content of the isochore (Equation 3.3, E_{iC}: expression level of isochore i normalised for the *GC* content of the isochore, GC_i: the *GC* content of isochore i).

$$E_{iC} = \frac{R_i}{R_t \times L_i \times GC_i} \times f = \frac{E_{iL}}{GC_i} \tag{3.3}$$

3.2.3 Expression level of genes

To investigate the expression at the gene level, the coding sequences (CDSs) for the mouse were retrieved from the NCBI Consensus Coding Sequence Database (CCDS) [166]. The CDSs were assigned to the isochores containing them, based on the coordinates of their exons as recorded in the CCDS database. Similarly to the expression level for the isochores, gene expression was measured with Equation 3.4 ($E_{g\ell}$: expression of gene g normalised for coding sequence length, R_g: exonic read count for gene g, R_{et}: total exonic read count for the tissue, ℓ_g: length of coding sequence for gene g, f': scaling factor). The scaling factor used here (10^{10}) is not the same as the one used for the isochoric expression.

$$E_{g\ell} = \frac{R_g}{R_{et} \times \ell_g} \times f' \tag{3.4}$$

3.3 Results and Discussion

The results of the alignment are presented in Table 3.1. Starting with a similar number of reads for each of the three adult tissues, half of those reads were successfully mapped for the brain and muscle and a third of the reads were successfully mapped for the liver. Out of the mapped reads, roughly half were mapped to known coding sequences for all three adult tissues. The samples for the immature brain tissues were considerably smaller, but the proportion of reads that were successfully aligned was similar to the adult issues. However, the proportion of reads aligned to known coding sequences was much lower, but consistent between the two immature tissues. This means either that the sequenced RNAs did not come from known coding sequences, or that the relevant genes were not included in the reference list of coding sequences. Coding sequences were obtained from the NCBI CCDS archive [167], whose mission is to compile a core list of well–annotated protein coding sequences. Thus it is most likely that the high number of reads that did not map to known coding sequences reflects genes that are more poorly understood and were not included

Tissue	Reads		
	Total	Aligned	Aligned to coding
adult Brain	31,116,663	14,219,266	6,635,861
adult Liver	31,578,097	11,353,537	6,449,293
adult Muscle	31,763,031	14,447,075	7,931,718
E18 Brain	2,956,444	1,643,644	223,398
P7 Brain	3,619,970	1,732,507	282,628

Table 3.1: Total number of reads in the dataset, number of reads successfully aligned, and number of reads aligned to known CCDS coding sequences

in the list as well as genes whose final product is the RNA instead of some protein.

As a consequence of the lower read coverage for the two immature tissues, the expression level measured at areas with few or no aligned reads is likely less reliable for these two tissues, so more emphasis will be placed on the peaks in expression, rather than the valleys. Therefore, some of the analyses, especially those focusing on the expression of genes, were limited to the three adult tissues only.

However, the overall picture of transcription presented in the transcriptome map attached in the Appendix, shows that the two immature tissues are in broad agreement with each other, as well as with the adult brain, for which the available reads were more abundant by one order of magnitude. Interesting differences can still be found in areas of high read coverage, where such observations are more reliable.

It is important to clarify that the expression levels discussed in the following sections are all relative to their respective tissue. Direct comparison of absolute values of expression between any of the five tissues is not possible, as each tissue represents an independent sequencing run and, therefore, differences in abolute values could be caused by a number of external variables. Thus, what will be discussed in this chapter is differences in the compositional distribution of expression, as opposed to absolute changes in the expression levels of any particular region.

3.3.1 The relationship between the base composition and the expression level

Using Equation 3.1, the expression profile of the isochores was plotted along the mouse genome, generating the first transcriptome map focused on isochores. This transcriptome map is presented in the Appendix Figures A.1 through A.21, separately for the three adult tissues (brain, liver, muscle) and the three maturity stages of the brain (embryonic day 18, post-natal day 7, adult).

It is apparent from the transcriptome map that the isochore expression levels roughly follow the respective GC levels of the isochores for all 5 tissues. It is also apparent that in most areas of the genome, the different tissues or developmental stages present similar or even identical expression levels. There are, however, regions where the expression levels differ in various possible ways:

- One adult tissue presents different expression level from the other two adult tissues: for example, the liver–specific peak at isochore 108 of chromosome 8 and the muscle–specific peak at isochore 125 (Fig. A.8), or the brain–specific peak at isochore 37 of chromosome 11 (Fig. A.11).

- Each of the three adult tissues presents a different expression level: for example, at isochore 70 on chromosome 5 (Fig. A.5) the liver is over-expressed, the muscle is under-expressed and the brain presents an intermediate level of expression. Interestingly, most of the large three-way differences in expression levels present the same tissue order as this example, but there are examples of different order as well, such as isochores 107 and 108 on chromosome 12 (Fig. A.12) and isochore 5 on chromosome 16 (Fig. A.16).

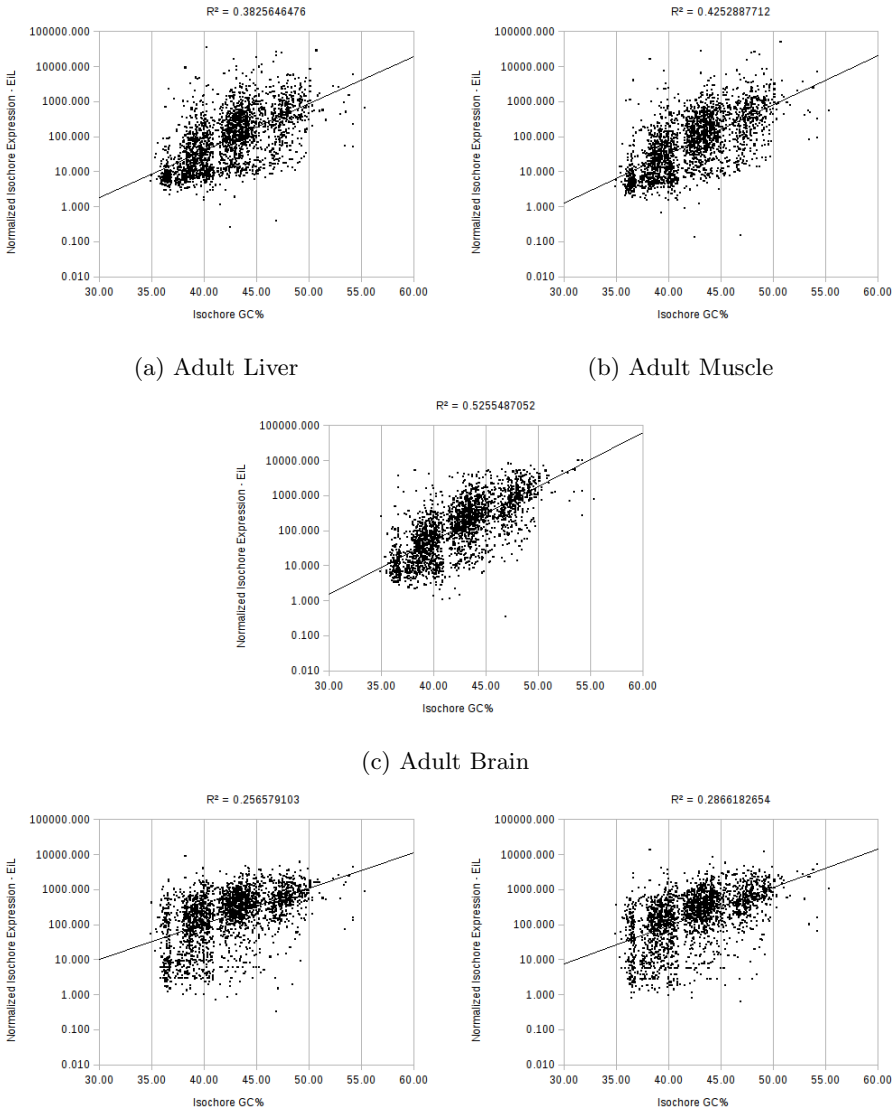
- The three adult tissues present identical expression, which is different from the expression of the embryonic or post-natal brain (for example on chromosome 16 the entire range between roughly isochores 60 and 80, Fig. A.16). It is also possible to witness the progressive change in expression from embryonic to adult brain (for example isochore 18 on chromo-

some 4, Fig. A.4), whereas in other cases the change between immature and adult likely occurs at a later stage not covered by the data here (for example isochores 12 and 62 on chromosome 4, Fig. A.4).

- Finally, it is possible to observe combinations of the possibilities already listed here, for example at isochore 70 on chromosome 5 (Fig. A.5), where the three adult tissues present very different expression levels, with the adult brain being relatively highly active, whereas at the embryonic and post-natal stages this region does not appear to be very active in the brain and resembles, instead, the expression level measured for the adult muscle.

The isochoric transcription map provides the locations of several such individual examples that mark areas where different tissues or different developmental stages present different expressional behaviour. As the data for the embryonic and post-natal brain came from a source independent to the source of the data for the three adult tissues, the fact that the expression profiles are very similar and even identical over large stretches of the genome serves as validation of these profiles and lends credibility to the observed differences in expression.

To quantify the relationship between base composition and expression at the isochore level, the covariation of these two variables was plotted for each tissue, revealing a strong positive exponential correlation between the *GC* content of the isochores and their normalised expression levels (Fig. 3.1), in agreement with previous reports [135, 157]. Interestingly, in all three adult tissues, most isochores in the *GC*-poor families (L1, L2) appear to be near or below the trend line of expression. On the contrary, in the two immature stages of the brain, many *GC*-poor isochores present considerable expression levels, comparable to those of *GC*-rich isochores. Such drastic differences are not observed for *GC*-rich isochores. This highlights a developmental stage–specific differential behaviour of *GC*-poor isochores, suggesting that L1 and L2 isochores are enriched for genes that are active early in an individual's life and become down-regulated or silenced as the individual matures. This had been hinted at by



(a) Adult Liver

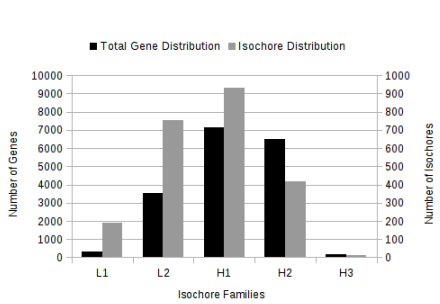
(b) Adult Muscle

(c) Adult Brain

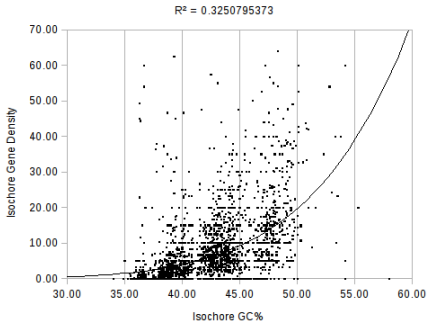
(d) Embryonic day 18 Brain

(e) Post-natal day 7 Brain

Figure 3.1: Covariation between the *GC* content of the isochores and their normalised expression level (E_{iL}, Equation 3.1). An exponential trend line is fitted to the data.



(a)



(b)

Figure 3.2: (a) Distribution of isochores to families and distribution of genes to the same isochore families. (b) Covariation between the *GC* content of isochores and their gene densities.

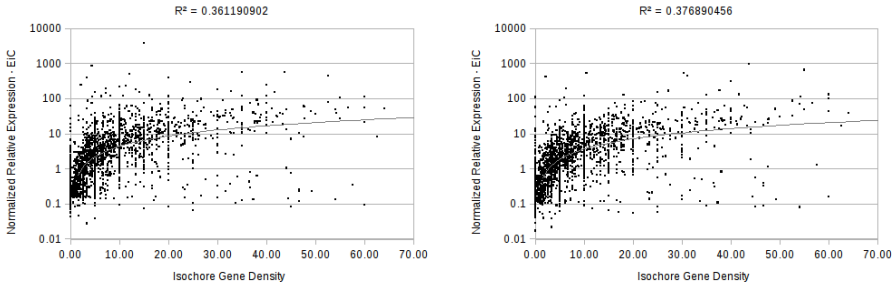
studies identifying highly conserved regulatory elements tied to developmental genes and located in "genome desert" (*GC*-poor) regions of vertebrates from fish to mammals [132, 168, 169], as well as by a study finding that developmental genes favour *AT*–ending codons [133], which are more common in *GC*-poor regions. These lines of evidence were often indirect and insufficient, but adding the present work's results from genome-wide expression data makes it clear that *GC*-poor regions are indeed largely implicated in developmental processes. This does not mean that all genes in *GC*-poor isochores are connected to development, nor that all developmental genes are located there. It also does not mean that all *GC*-poor isochores are implicated in development, as the Figure 3.1 clearly shows a large proportion of L1 and L2 isochores showing low overall expression even at the developmental stages. This last observation could be overturned in the future by analysing more time points along development as well as more tissues.

However, it is known that higher *GC* content also coincides with higher gene density [118, 124–126] (Fig. 3.2), and, in turn, the higher concentration of genes being expressed can be responsible for the increased expression levels of a genomic region. This alone might explain the correlation between the expression level and the base composition. Figure 3.3 demonstrates that, indeed, gene

density accounts for a large part of the expression levels and their correlation with *GC* content.

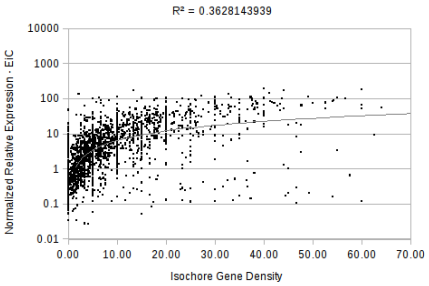
To remove the influence of the gene content of the isochores and reveal any potential direct connection between the base composition and the expression level, the expression levels were further normalised for the number of genes in each isochore (Equation 3.2). Plotting the covariation of this expression against the *GC* content of the isochores, reveals a persisting positive exponential correlation between the *GC* content and the expression level of the isochores in the adult tissues (Fig. 3.4). Therefore the higher gene density of the *GC*-richer isochores is not entirely responsible for the higher expression levels and other properties must factor into the expression level at the regional level. Possible mechanisms through which the base composition can directly affect the expression levels may be the presence of different regulatory sequence elements than in *GC*-rich regions [137, 138], or differences in methylation levels and chromatin structure [132, 140].

However, the persisting correlation between *GC* content and expression seen in the three adult tissues does not also apply to the two developmental stages of the brain. There, the correlation is eliminated when the influence of gene density is factored out, indicating that in early developmental stages, all isochores are similarly active, regardless of their composition. In combination with the observation that the L1 and L2 isochores later become much less active in the adult tissues, this reinforces the suggestion that regulatory mechanisms are acting at the genomic level, such as changes in chromatin compaction patterns. Such changes have, indeed, previously been reported [132, 140]. Alternatively, changes in the concentrations of transcription factors in conjunction with the different regulatory elements between *GC*-poor and *GC*-rich isochores could also produce this effect.

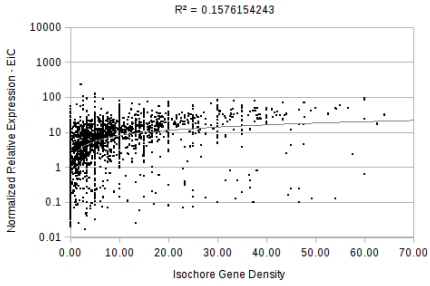


(a) Adult Liver

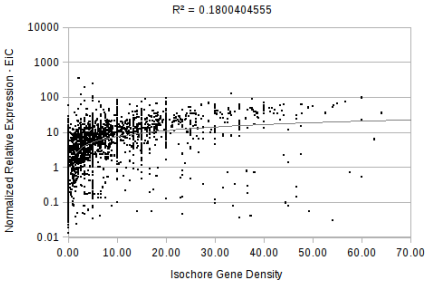
(b) Adult Muscle



(c) Adult Brain

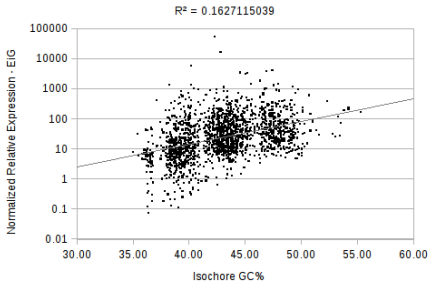


(d) Embryonic day 18 Brain

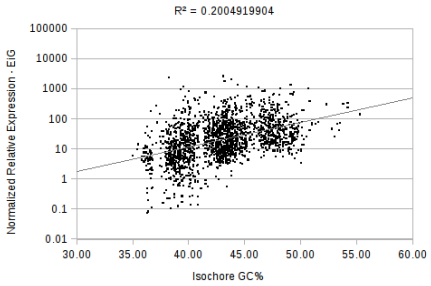


(e) Post-natal day 7 Brain

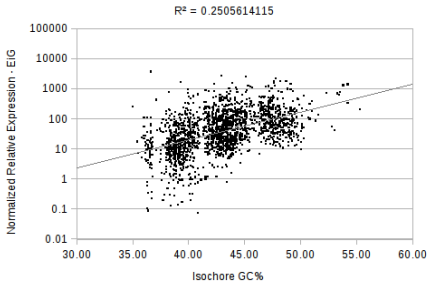
Figure 3.3: Covariation between the gene density of the isochores and their expression level normalised for GC content (E_{iC}, Equation 3.3). A power–law trend line is fitted to the data.



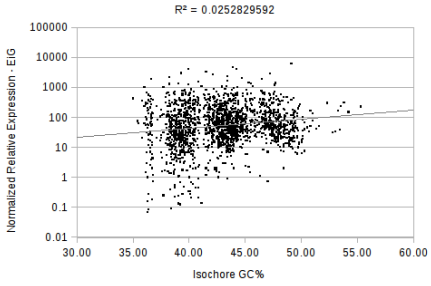
(a) Adult Liver



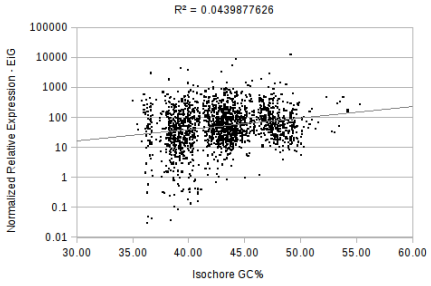
(b) Adult Muscle



(c) Adult Brain



(d) Embryonic day 18 Brain



(e) Post-natal day 7 Brain

Figure 3.4: Correlation between the GC content of the isochores and their normalised expression level, after gene density has been taken into account (E_{iG}, Equation 3.2).

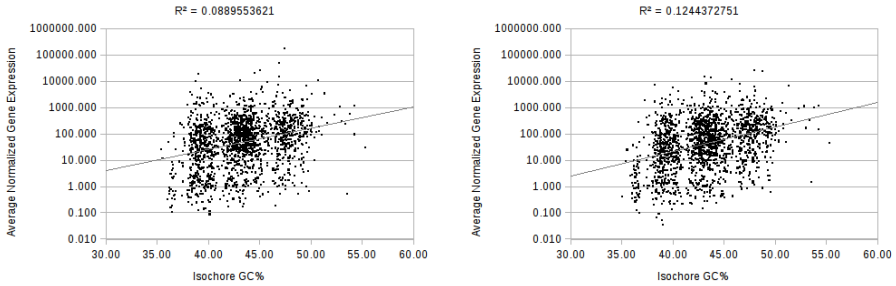
3.3.2 Gene expression

The transcriptome maps presented revealed localised differences in the expression levels between different tissues and different developmental stages. These differences were shown to follow certain trends at the level of isochores, as discussed in the previous section. In order to attribute the differences at the isochore level to differences at the gene level, the expression data was mapped onto the NCBI Consensus Coding Sequence Database (CCDS) and normalised for the length of each coding sequence (CDS) and the total number of reads mapped onto the coding sequences of the tissue (Equation 3.4). Figure 3.5 demonstrates that the normalised expression of genes, averaged per isochore, follows similar patterns as those shown for the expression at isochore level, after the gene density was factored out (Fig. 3.4), albeit the correlations are considerably weaker.

Here again, despite the adult tissues' showing a positive correlation between the gene expression and the *GC* content of the isochores, the gene expression during development appears to be independent of the base composition of the isochores, with genes in *GC*-poor isochores matching in expression level the genes in *GC*-rich isochores. Therefore the positive correlation between the expression level and the *GC* content observed in the adult tissues is the result of differential regulation of genes in *GC*-poor and *GC*-rich isochores, possibly at the level of entire isochores, as discussed previously.

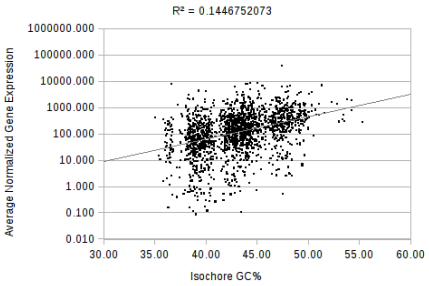
Indeed, the fraction of genes, per isochore family, that are not expressed in any of the three adult tissues is highest for the *GC*-poorest isochores and lowest for the *GC*-richest ones (Fig. 3.6). This is in contrast with the above observation that genes in *GC*-poor isochores are expressed with similar intensity as the genes in *GC*-rich isochores and means that L1 and L2 isochores contain mostly genes that are only active early in development and are silenced later in the individual's life. On the contrary, Figure 3.6 shows that the vast majority of the genes in *GC*-rich isochores are active in at least one of the adult tissues.

Aside from the connection between development and base composition al-

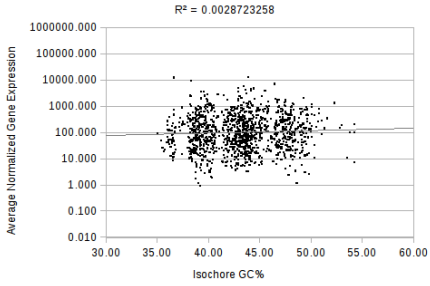


(a) Adult Liver

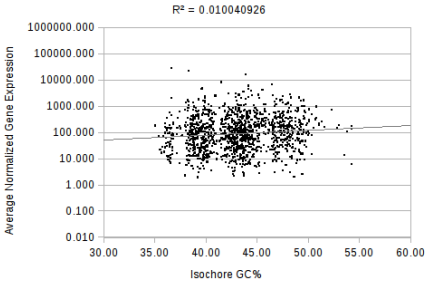
(b) Adult Muscle



(c) Adult Brain



(d) Embryonic day 18 Brain



(e) Post-natal day 7 Brain

Figure 3.5: Correlation between the *GC* content of the isochores and the average normalised expression level of the genes located in them ($E_{g\ell}$, Equation 3.4). An exponential trend line has been fitted to the data.

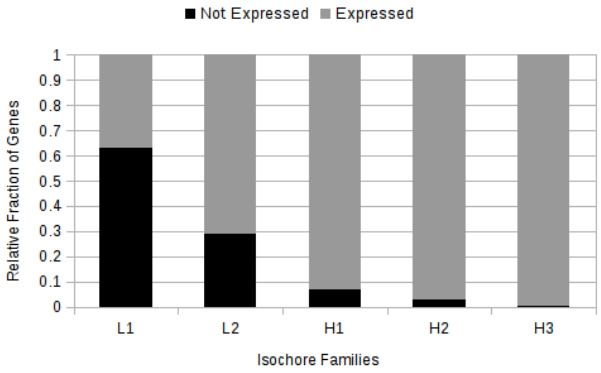
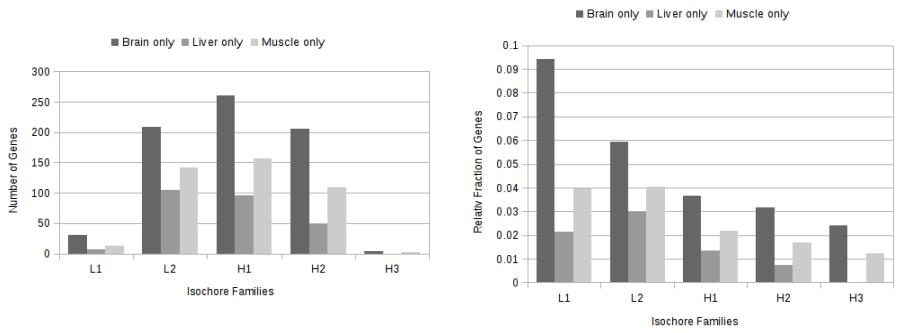


Figure 3.6: Distribution of genes not expressed in any of the three adult tissues, shown as a fraction of the genes located in each isochore family.

ready presented, *GC*-poor isochores also show a connection with tissue–specific genes. As Figure 3.7 shows, in absolute numbers, these tissue–specific genes follow similar distributions as the total number of genes (Fig. 3.2) and are spread across all isochore families, in agreement with previous reports [135]. However, when seen as fractions of the genes located in each isochore family, a larger fraction of genes in L1 and L2 isochores is tissue–specific than in *GC*-rich isochores. It must be noted, though, that the number of such genes may be overestimated, due to the limited number of tissues involved in the study and the fact that the results refer actually more to entire organs than specific specialized tissues, thus some tissue overlap should exist between the three adult samples. Conversely, there may also be genes specific to other tissues not included in this study, which currently falsely appear not to be expressed in any tissue and cause an underestimation of the number of tissue–specific genes. I do not expect the overall distribution to change dramatically, but it would be interesting nonetheless to see how additional data from other organs as well as from more precisely isolated tissues compares to these results, in order to corroborate the observations made here.



(a) (b)

Figure 3.7: Distribution of genes expressed in only one of the three adult tissues, shown in (a) absolute numbers and (b) as a fraction of the genes located in each isochore family.

3.4 Conclusions and perspectives

In this chapter, the read alignment algorithm *REAL* was used to study the expression patterns of isochores and to add to the evidence linking gene expression to the local base composition. To this end, the first transcriptome map of the mouse focused on isochores was produced, for three distinct tissues from adult individuals (brain, liver, muscle) and for two developmental stages of one of these tissues (brain at embryonic day 18 and post-natal day 7) [170, 171]. It is also the first time NGS data has been used for this purpose.

The connection between composition and expression has been criticised and questioned, as different methodologies have lead to varying and conflicting results [123, 126, 135, 153, 157]. Much of the controversy stems from the methodologies used [157]. This study used whole–genome RNA-seq data, which is a more comprehensive and objective method than previously used ones and adds conclusive evidence that there is in fact a positive correlation between the isochore composition and the expression level of both genes and entire isochores, as well as a subtle but definite connection between isochore composition and the breadth of gene expression and also between isochore composition and the developmental timing of expression. It has been proposed that the compart-

mentalization of the genome into isochores acts as a large–scale gene regulation mechanism acting at the level of chromatin structure [141, 143]. This hypothesis offers a plausible and attractive mechanism for the regulation of collocated groups of genes that are not required in all tissues or at all stages of an organism's or cell's life. The results of this study can be explained by this hypothesis, but other explanations may also apply.

This work also highlights that the correlation between expression and composition is complex and at least partly indirect, with other features that correlate with the composition also being responsible, such as the gene density. In fact, the compartmentalization of the genome into isochores correlates with a large number of features, many of which can affect expression (methylation patterns, different regulatory elements, different functional roles of the genes, chromatin structure), as discussed already, and the causative relationships connecting all these features with one another remain unclear. Thus, more research needs to be directed in deducing how isochores evolved and, more crucially, how the various correlated features of isochores evolved. Isochores are usually thought of primarily as a compositional compartmentalization, because this is the feature that led to their discovery, but it is possible that one of the other features is the basis of the compartmentalization and the underlying cause of the observed features and correlations.

Chapter 4

Functional Effect Classification for Single Nucleotide Polymorphisms

4.1 Background

DNA sequencing has entered a new era, in which high throughput technologies enable large amounts of sequence data to be generated quickly and at low cost. Much of the data generated is aimed at the discovery of disease causing genetic variation. A key challenge in this process is the interpretation of the functional consequences of variant alleles. Given the extensive number of variants identified by whole genome or exome re-sequencing studies, it is infeasible to interrogate the functional consequences of all the variant alleles at all the gene loci experimentally.

A number of bioinformatics solutions for the annotation, scoring and classification of variants have been developed to address this challenge (Table 4.1) and several comprehensive overviews of the available tools and methods have been carried out [172–175]. Such tools are providing a supportive role in the exper-

imental validation of disease-related alleles, by prioritising candidate variants with predicted functional consequences as causes of specific inherited diseases and cancers. These bioinformatics approaches draw from a broad range of existing knowledge about the structure, function, conservation and annotation of the genes, transcripts and proteins in which the variants are located. As will become apparent in the following sections, some of these tools represent fully fledged strategies, whereas others depend to various extents on pre-existing predictors and classifiers, either by extending previous work by the same authors, or by combining multiple independent tools into a single verdict.

Despite the wide variety and availability of individual classification tools, no attempt had been made to study the potential accuracy of consensus strategies by the time this work was coneived. Thus, in this chapter, my aim was to identify the best method with which to prioritise non-synonymous substitutions as candidate causes of monogenic diseases. I assessed the performance of nine existing tools that draw on different resources to classify non-synonymous single nucleotide substitutions as likely or unlikely to have a serious impact on a protein's function. I also propose my own consensus strategy. The tools evaluated here are SIFT [76], PolyPhen2 [77], MutPred [78], SNPs&GO [79], PANTHER [80], PhD-SNP [81] and Mutation Assessor [82], as well as the consensus classifiers Condel [92] and CAROL [84], both of which appeared around the same time as the results of this study. Together, these tools encompass a broad range of non-synonymous DNA sequence variant classification criteria and methods. Each has been independently evaluated previously [92, 175], but never together on a common dataset.

In order to justify this selection, it is first necessary to review the features of each tool and its applicability to the problem at hand, encompassing information retrieved from the original publications, the respective web-pages and the reviews.

Tool	*Ref*	*Year*	*Coding*	*Non-coding*	*Classifier Type*
SIFT	[76, 176, 177]	2001	X		mathematical
PolyPhen	[178]	2002	X		DT
PANTHER	[80, 179, 180]	2003	X		HMM
PupaSNP Finder	[181]	2004		X	
nsSNPAnalyzer	[83]	2005	X		RF
Pmut	[91]	2005	X		NN
PupasView	[182]	2005	X	X	Pmut, PupaSNP
LS-SNP	[95]	2005	X		SVM
SNPEffect	[183, 184]	2005	X		
PhD-SNP	[81]	2006	X		SVM
PupaSuite	[89]	2006	X	X	
FASTSNP	[101]	2006	X	X	DT
SNPs3D	[102, 185]	2006	X		SVM
SNAP	[86]	2007	X		NN
SNPnexus	[88]	2008	X	X	
F-SNP	[96]	2008	X	X	DT
SNPs&GO	[79]	2009	X		SVM
MutPred	[78]	2009	X		RF
PolyPhen2	[77]	2010	X		Bayesian
PON-P	[99]	2009	X		
Ensembl VeP	[97]	2010	X	X	
SNPs&GO3D	[87]	2011	X		SVM
Mutation Assessor	[82]	2011	X		mathematical
Condel	[92]	2011	X		mathematical (SIFT, PolyPhen2, M. Assessor)
CAROL	[84]	2012	X	X	mathematical (SIFT, PolyPhen2)

Table 4.1: List of SNP effect classification tools and their approximate release year. (RF: random forests, SVM: support vector machines, HMM: hidden Markov models, NN: neural networks, DT: decision trees)

4.1.1 Features based on the amino acid sequence and protein annotation

In order to evaluate the effect of coding variants, most classification and prediction tools rely heavily on properties and features extracted at the protein level. Proteins are generally better understood than nucleic acids and more is known about the mechanisms by which substitutions affect their structure or function. Also, proteins undertake nearly all the tasks in cells and are the agents that carry out the function of most genes, therefore, turning to them for answers is a natural step in understanding how some DNA substitutions may affect phenotype.

Simple considerations involve the type of the amino acid residue affected [79, 81, 83, 87, 97, 101] and the amino acid sequence flanking the variant position [79, 81, 86, 87]. The predicted secondary structure is also considered by many of the tools [77, 78, 86, 89, 91, 178], as is the prediction of potential transmembrane segments [77, 78, 89, 91, 178, 184], whereas protein targeting and subcellular localisation is considered by few [178, 184]. Finally, information about the function of the variant position and the domain or protein it is located in are also criteria in some methods [78,178,184], along with prediction of the variant's effect on protein stability, solubility and aggregation tendencies [78, 86, 91, 184].

A summary of the amino acid sequence properties and protein annotation used in the evaluation of coding variants, as well as the resources employed to that end by each tool, is presented in Table 4.2.

4.1.2 Features based on the protein structure

Although a lot of information can be derived with only the amino acid sequence available, it seems intuitive that examniation of the actual 3D structure of a protein would give more accurate results than the examination of its predicted structure. Indeed, properties that are considered by the prediction and classification tools are the protein's actual secondary structure

Tool	*Residue and Flanking Sequences*	*Predicted Structure*	*Function*	*Sub-cellular Localisation and Membrane Topology*	*Predicted Stability and Solubility*
PolyPhen		Uniprot, coils2	Uniprot	Uniprot, TMHMM, SignalP	
nsSNP Analyzer	residue type				
Pmut		PROFphd		PROFphd	PROFphd
SNPEffect 4.0			Catalytic Site Atlas, Pfam, SMART, LIMBO, lipid anchoring	PSORT, TMHMM	TANGO, WALTZ
PhD-SNP	substitution type and flanking residue types				
PupaSuite		PROFphd	SNPEffect	PROFphd, SNPEffect	PROFphd, SNPEffect
FASTSNP	residue type				
SNAP	sequence environment	PROFphd			PROFphd
F-SNP		PolyPhen, SNPEffect	PolyPhen, SNPEffect	PolyPhen, SNPEffect	SNPEffect
SNPs&GO	PhD-SNP				
MutPred		PROFphd, MARCOIL	yes	Phobius	PROFphd, PONDR, i-Mutant2
PolyPhen2		coils2	Uniprot	Uniprot, TMHMM	
PON-P	PhD-SNP, SNAP	SNAP, PolyPhen2	PolyPhen2	PolyPhen2	SNAP
Ensembl VeP	residue type	PolyPhen2	PolyPhen2	PolyPhen2	
SNPs&GO 3D	PhD-SNP				
Condel		PolyPhen2	PolyPhen2	PolyPhen2	
CAROL		PolyPhen2	PolyPhen2	PolyPhen2	

Table 4.2: Variant features extracted from the amino acid sequence, annotation and properties prediction resources.

[77, 83, 95, 184], assessment of local steric constraints affected by the amino acid change [77, 87, 95, 102, 178]) and the overall change in structure flexibility and stability [77, 78, 95, 102, 178, 184]. Also considered are changes to the protein's charge and solvent accessibility [77, 83, 87, 95, 102, 178] and inter-molecular and intramolecular interactions [77, 95, 102, 178].

The drawback of this approach is that 3D structures are available for relatively few proteins. To circumvent this issue, several tools employ homology modelling [95, 102, 178, 184], which allows them to use the structure of a closely similar protein, when one such exists. However, obtaining meaningful results from homology modelling depends greatly on the similarity of the chosen homologous sequences and their evolutionary distance from the query. Results from nearly identical sequences from closely related species are likely to be trustworthy, but the same cannot be said for homologs from more distant species or for paralogs from any species. Of course, homology modelling is of no use when no suitable close homolog exists, which is quite a common problem as sequence databases grow at a pace much faster than structure databases.

Table 4.3 summarises the structural properties of proteins that are taken into account when evaluating the effect of coding variants and notes the various resources employed by the different tools.

4.1.3 Features based on amino acid sequence homology

Aside from evaluating an amino acid substitution using properties derived from the amino acid sequence, structure and annotation of a protein, sequence conservation has proven a powerful predictor of how well tolerated a variant may be [176]. The main principle behind this is that residues that are highly conserved among homologs of a protein are likely to be important for that protein's function or structure, therefore their substitution is more likely to be deleterious. On the other hand, residues that show a greater degree of variability among homologs are likely to be less crucial to the structure or function, therefore their substitutions are more likely to be tolerated.

Tool	*Secondary Structure*	*Steric Constraints*	*Charge, Solubility*	*Flexibility, Stability*	*Interaction, Function*
PolyPhen		DSSP	yes, DSSP	yes	yes, PDB, HBPlus
nsSNP Analyzer	STRIDE		yes, ENVIRONMENT		
LS-SNP	DSSP	yes	DSSP, PolyPhen	yes	MODBASE
SNPEffect 4.0	FoldX			FoldX	
PupaSuite				SNPEffect	
SNPs3D		yes, PROCHECK	yes	yes	yes
F-SNP	LS-SNP, SNPEffect	LS-SNP, PolyPhen, SNPs3D	LS-SNP, PolyPhen, SNPs3D	LS-SNP, PolyPhen, SNPEffect	LS-SNP, PolyPhen, SNPs3D
MutPred				i-Mutant	
PolyPhen2	DSSP	yes, DSSP	yes, DSSP	yes	PDB
PON-P	PolyPhen2	PolyPhen2	PolyPhen2	PolyPhen2, i-Mutant	PolyPhen2
SNPs&GO 3D		yes	DSSP		
Condel	PolyPhen2	PolyPhen2	PolyPhen2	PolyPhen2	PolyPhen2
CAROL	PolyPhen2	PolyPhen2	PolyPhen2	PolyPhen2	PolyPhen2

Table 4.3: Variant features extracted from the protein structure, annotation and properties prediction resources.

Several tools exploit conservation information, either by explicitly building the multiple sequence alignments for a query protein and measuring the level of conservation at the variant position [76, 77, 79, 81, 82, 95, 102, 178], or by exploiting pre-existing curated multiple sequence alignments, usually in the form of a Hidden Markov Models database [77–80, 86, 89].

The tools making use of sequence conservation information and the resources they use are presented in Table 4.4.

4.1.4 Features based on the nucleotide sequence and annotation

Examination of the variant's properties at the protein level potentially offers a multitude of information about the effect of the variant on the structure and function of the protein. There are, however, a number of tools that also evaluate variants using properties and annotation at the level of the nucleotide sequence. This allows them to expand their detection range to include variants that may affect a gene's function at a stage prior to the translation into protein. These variants are generally not coding for amino acids, but are part of regulation elements that control gene transcription [77, 88, 89, 96, 97, 99, 101] or maturation and editing of the primary transcript [88, 96, 97, 101]. Similarly to the protein–based features, nucleic acid–based feature combine a mix of annotation about the presence of known functional elements, modifications sites and sequence conservation.

A summary of the use of gene-level properties and annotation by the classification tools is presented in Table 4.5, along with the resources from which such information is extracted.

Tool	*Residue Frequencies*	*Substitution Frequencies*	*Sequence Retrieval and Alignment*
SIFT	yes		PSI-BLAST, MOTIF
PolyPhen	yes(PSIC)		BLAST
PANTHER	HMM		HMM
PupaSNP Finder	InterPro		
nsSNPAnalyzer	SIFT		
Pmut			PSI-BLAST
PupasView	Pmut		
LS-SNP	yes		PSI-BLAST, SAM-T2K
PhD-SNP	yes		BLAST
PupaSuite	PupaSNP Finder, Pmut		
SNPs3D	yes	yes	PSI-BLAST
SNAP	PSIC, HMM(Pfam)	yes	PSI-BLAST
F-SNP	SIFT, PolyPhen, LS-SNP, SNPs3D	SNPs3D	
SNPs&GO	PhD-SNP, PANTHER		
MutPred	SIFT, HMM(Pfam)	yes	PSI-BLAST
PolyPhen2	PSIC, HMM(Pfam)		BLAST, LEON, clustpack, MAFFT
PON-P	PhD-SNP, SNAP, PolyPhen2		
Ensembl VeP	SIFT, PolyPhen2		
SNPs&GO3D	PhD-SNP	PhD-SNP	
Mutation Assessor	yes	yes	BLAST, MUSCLE
Condel	SIFT, PolyPhen2, M.Assessor	M.Assessor	
CAROL	SIFT, PolyPhen2		

Table 4.4: Variant features extracted from multiple sequence alignment.

Tool	*Sequence Conservation*	*Promoters, Enhancers, Splicing*	*Untranslated Regions, Post-translational Modifications*	*Other*
PupaSNP Finder		TRANSFAC, Ensembl		GO terms
PupasView		PupaSNP Finder		PupaSNP Finder
PupaSuite	yes, Ensembl	PupaSNP Finder, ORegAnno, JASPAR		PupaSNP Finder
FASTSNP		TFSearch, Uniprot, ESEFinder, RescueESE, FAS-ESS	yes	
SNPs3D				KEGG, HGMD, dbSNP
SNPnexus		Ensembl, NCBI, UCSC Gen. Brows., Ace View, miRBase, TFSearch, FirstEF	Ensembl, NCBI, UCSC Gen. Brows.	Genet. Assoc. DB, HapMap
F-SNP	UCSC Gen. Brows.	Ensembl, UCSC Gen. Brows., TFSearch, Consite, ESEFinder, RescueESE, ESRSearch, PESX	OGPET, Sulfinator, KinasePhos	SNPs3D
SNPs&GO				GO terms
PolyPhen2		yes(CpG islands)		
PON-P		PolyPhen2		
Ensembl VeP		Ensembl, UCSC Gen.Brows., TFSearch, Consite	Ensembl	
SNPs&GO3D				SNPs&GO
Condel		PolyPhen2		
CAROL		PolyPhen2		

Table 4.5: Variant features extracted from the nucleotide sequence and annotation.

4.2 Materials and Methods

4.2.1 Tool selection and execution

Seven individual tools and two consensus tools for predicting the functional consequences of non-synonymous DNA sequence variation were selected for comparison. The selection criteria were availability, ease of use and ability to evaluate large numbers of variants through either batch queries or automated submission of individual variants.

The task of obtaining functional effect predictions from multiple tools can be simplified with the use of meta-tools such as PON-P [99] and the Ensembl VeP [97], both of which serve as gateways to a multitude of bioinformatics resources relevant to the functional study of variants, including several of the tools selected for this study (SIFT, PolyPhen2, SNPs&GO, PhD-SNP). However, using each individual tool's own interface allows for better control over the parameters and data submitted in the queries and simplifies retrieval of the original results.

SIFT and PolyPhen2 support batch queries online on their respective websites. PhD-SNP, SNPs&GO and Mutation Assessor have limited or no support for batches, but their web-APIs can be queried in an automated way. Thus, custom scripts, written in PERL, were implemented for each of the three tools, in order to submit variants one by one. MutPred has limited batch support via its website, but was kindly executed locally by its authors on my behalf upon request. PANTHER has limited online batch support, or alternatively requires an extensive and complicated local installation of the database. However, one of the other selected tools, SNPs&GO, queries PANTHER internally and displays PANTHER's classification result when available. Therefore, PANTHER results were entrusted to SNPs&GO. This is the only exception to the rule of using each tool's own interface. Condel is available both online and as a local PERL script, the latter supporting batches and requiring no installation. The script originally required pre-obtained scores from five independent methods, whereas the online version employs only three of them, all three of which are among the

seven tools chosen for this study (SIFT, PolyPhen2 and Mutation Assessor). Thus, the script was modified so that the consensus prediction would be based on these three tools only, emulating its online counterpart. CAROL is available as a simple script in the *R* mathematical language.

With regards to input format, all tools accepted variants in the format of amino acid substitutions paired with an identifier code for the respective protein. SIFT and PolyPhen2 also accept queries in the format of nucleotide substitutions paired with the corresponding genomic coordinates. Both are able to process coordinates from the latest two human genome assemblies (NCBI36/hg18 and NCBI37/hg19), therefore they were executed additional times to explore the influence of using nucleotide substitutions instead of amino acid substitutions. Finally, PolyPhen2, offers two versions of its classifier, trained on different datasets [77]: The HumVar set is composed from variants annotated in Uniprot as disease-associated and ones without such annotation, whereas the HumDiv set is more strict and is composed from only those variants annotated to cause Mendelian diseases and variants determined to be neutral by evolutionary comparison to the corresponding proteins of other vertebrates. Both versions of the classifier were tested.

4.2.2 Benchmarking data sets

In order to evaluate the prediction tools, a sufficiently large dataset of positive (affecting function, causing disease) and negative (not affecting function, neutral) variants was necessary. Two sources of human non-synonymous variant data were selected; one that is enriched for variants with experimentally confirmed functional consequences and a second variant dataset likely to contain a reduced level of functional variation. The set of DNA variants with functional consequences comprise variants previously implicated in the pathogenesis of inherited human disease and were extracted from the commercial catalogue HGMD Pro v.2011.1 [186]. The set of putative neutral variation was selected from variants identified by the 1000 Genomes Project Pilot Project [3] (release

July 2010). An additional validation set was extracted from PhenCode [100]. The data extraction was executed via a series of custom PERL scripts applied to the downloaded plain-text versions of the databases.

As the different tools benchmarked here use different databases as their sources for annotation and amino acid sequences, it was necessary to also build a cross-referencing index containing the Uniprot, NCBI_transcript, NCBI_peptide, ENSEMBL-protein and ENSEMBL-transcript codes, in order to enable the retrieval of all the relevant identification codes for each variant.

Negative cases

Unlike disease-associated variants, for which annotation in various databases exists, neutral variants are harder to collect with confidence. The problem arises from the fact that lack of annotation as damaging does not guarantee neutrality. Uniprot [187] lists some variants that are annotated as non-damaging and has been used for this purpose in other studies [78, 79, 81–83, 92]. An alternative option is to assume neutrality based on sequence conservation; changes observed in the same protein between humans and other mammals are assumed to be neutral [77, 91, 96, 102]. The third option is to assume that variations that are wide-spread in the human populations are unlikely to be harmful [84]. In this study, the third strategy was chosen and data was collected from the 1000 Genomes Project [3].

The 1000 Genomes Project is an effort to catalogue the naturally occurring genetic variation in humans. The pilot data is based on low coverage whole genome sequencing of 179 individuals, distributed in three groups with distinct geographic origin: African (YRI), Caucasian (CEU), East-Asian (CHBJPT). Variants detected in each geographic group were recorded separately: (YRI)–10,556,156, (CEU)–7,724,854, CHBJPT–6,107,825. In the present study, variants found with high allele frequency in each of the three populations separately are considered very likely to have no serious functional impact and are, thus, assumed to be adequately neutral.

Firstly, variants of very low (< 0.05) or very high (> 0.95) allele frequency were discarded from each of the three sets as either too rare or too established respectively. The three sets were then intersected, yielding $3,074,219$ variants with high frequency in each population. In order to isolate the ones that result in amino acid substitutions, they were annotated using Annovar [100]. Only $8,461$ of the variants were annotated as "non-synonymous". The amino acid sequences for them were obtained from the NCBI RefSeq, using the NCBI_transcript codes assigned by Annovar. Variants for which no amino acid sequence could be retrieved, or for which sequence conflicts arose, were discarded. Such conflicts would inhibit the benchmarked tools from mapping the variant to the correct amino acid sequence and correct amino acid substitution and were thus not suitable for the purpose of this study. Variants were also discarded when no corresponding Uniprot accession number could be found, as the accession numbers are required by some of the tools.

After all processing, our putatively neutral data set from the 1000 Genomes Project comprised $7,791$ non-synonymous variants across $4,555$ genes — this is defined as the *negative set*.

Positive cases

Potential sources of disease-causing variants, as used in previous works, are mainly Uniprot [187] (used in [77, 79, 81–83, 87, 91, 92, 95]) and the The Human Gene Mutation Database (HGMD) [186] (used in [78, 84]). In this study, the latter was chosen due to ready availability of the larger commercial version.

HGMD Pro [186] is a commercial curated catalogue of disease-causing and disease-associated variants, collected from the literature. In the release version used here, $61,902$ single nucleotide substitutions linked to disease were listed. The database lists mutations from both monogenic (Mendelian) and complex diseases. In the latter case, the disease only occurs as a combination of multiple mutations, and its severity may vary depending on the type and number of mutations present. Such mutations comprise a grey zone and are not suitable

for building a confident set of disease-causing mutations. Therefore, 48,241 "disease mutations" (DM) were extracted, located on autosomal chromosomes.

As discussed in section 4.1.4, not all disease-causing mutations code for amino acids. Following annotation with Annovar, 38,358 non-synonymous variants were isolated. The corresponding amino acid sequences were obtained from HGMD, and the NCBI transcript codes were obtained from Annovar. Using the cross-referencing index mentioned in 4.2.2, the corresponding Uniprot accession numbers were obtained. This allowed the comparison of the sequence supplied by HGMD with the primary sequence listed in Uniprot. If the sequences were not identical, the variants were discarded, as this would cause problems for some tools. This discrepancy can happen as a result of alternative splicing, a process that causes different amino acid sequences to be coded for by the same gene.

This annotation and elimination process lead to a pool of 28,868 non synonymous disease-causing single nucleotide substitutions. In order to have similarly sized positive and negative sets and avoid the dataset composition bias in the performance evaluation, 7,800 disease variants were sampled semi-randomly. This was achieved by iterating through all the genes included in the disease variant pool and randomly picking one mutation from each gene in each iteration, until the target sample size was reached. The reason for not choosing variants completely at random from the pool was to maximise the diversity of proteins represented. The final set consists of 7,779 variants across 1,448 genes, collected in 10 iterations — this is defined as the *positive set*.

Genome assembly versions

All *positive* and *negative* variants were originally derived from the NCBI36/hg18 human genome assembly. In order to test the effect of using the newest genome assembly, all the coordinates were converted to the NCBI37/hg19 human genome assembly using the UCSC Genome Browser's liftOver utility[1] [188]. This is relevant only for SIFT and PolyPhen2, which accept nucleotide-based queries. All

[1] http://genome.ucsc.edu/cgi-bin/hgLiftOver

	Predicted Positive	Predicted Negative	Not Classified
Positive (P)	true positive (TP)	false negative (FN)	unknown (UP)
Negative (N)	false positive (FP)	true negative (TN)	unknown (UN)

Figure 4.1: Overview of the outcome possibilities for binary classification. In the ideal case where a predictor classifies all queries, then $UP = 0$ and $UN = 0$, thus $P = TP + FN$ and $N = TN + FP$. If the predictor is unable to classify some of the queries, then $P = TP + FN + UP$ and $N = TN + FP + UN$.

other tools require amino acid-based queries.

Validation data set

The source of the *positive set* used in this study, HGMD, has been used in the training of MutPred and poses a potential source for bias in favour of this tool. Therefore, an independent validation dataset was compiled. PhenCode [189] was chosen as the source for the validation set. PhenCode (downloaded in May 2013) listed 90,609 variants, of which 73,703 were coding (exonic) substitutions. The 20,470 coding variants annotated in PhenCode [189] as disease-associated were extracted and compared with the 61,902 single nucleotide variants in the HGMD Pro catalogue. All PhenCode substitutions with NCBI36/hg18 coordinates that were also listed in HGMD were discarded, in order to eliminate all potential overlap and ensure the dataset is independent. Following the cross-referencing to obtain protein and transcript codes for RefSeq and Ensembl, 13,142 disease-associated amino acid substitutions remained. Out of these, a subset of 5,000 was randomly sampled. Due to a discrepancy in PhenCode causing some variants to appear in duplicate, the final *validation set* consisted of 4,985 amino acid substitutions in 1,164 proteins.

4.2.3 Evaluation

The selected classification tools were evaluated with metrics applicable for binary classification problems. A balanced dataset of neutral and disease-causing single amino acid substitution variants, described in section 4.2.2, was used for the benchmarking of the tools. Performance was assessed using the counts

defined in Figure 4.1 to calculate the following metrics:

Definition 4.1. Sensitivity*: The proportion of* Positives *correctly classified as positive [190].*

$$Sens = \frac{TP}{TP + FN} \tag{4.1}$$

Definition 4.2. Specificity*: The proportion of* Negatives *correctly classified as negative [190].*

$$Spec = \frac{TN}{TN + FP} \tag{4.2}$$

Definition 4.3. Accuracy*: The proportion of correctly classified variants overall [190].*

$$Q2 = \frac{TP + TN}{TP + TN + FN + FP} \tag{4.3}$$

Definition 4.4. Matthews Correlation Coefficient*: A measure of correlation between observation and prediction [190].*

$$MCC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{4.4}$$

Definition 4.5. Proportion Classified*: The proportion of submitted queries that were successfully classified.*

$$PC = \frac{TP + FN + TN + FP}{P + N} \tag{4.5}$$

Definition 4.6. Area Under the ROC Curve (AUC)*: A measure of how much better a binary classifier is, compared to a purely random classification. [190].*

These binary classification metrics have one common limitation. Their standard definitions (Equations 4.1, 4.2, 4.3, 4.4) assume that classification is complete, that all the queries are classified. In practice, however, some predictors may fail, for various reasons, to classify some of the queries. In this case, these metrics suffer in that they cannot discriminate between predictors that classify all the queries and those that leave many queries unclassified. Therefore the proportion of queries classified (Equation 4.5) is always presented along with these

metrics, and must be taken into account when judging the tools' performance.

Another limitation of the metrics is that they only apply to binary classification. This is not by default the classification mode for several of the tools benchmarked here. Whilst SNPs&GO, PANTHER, PhD-SNP, Condel and CAROL offer indeed a simple binary prediction, SIFT and PolyPhen2 generate three ranked classes, Mutation Assessor generates five and MutPred none. In order for the metrics to become applicable, the ranked classes were merged down to simulate a binary classification. For SIFT and PolyPhen2, two scenarios were considered: (i) The intermediate class was considered to be neutral and (ii) it was considered to be damaging. In the case of Mutation Assessor, which employs five classes, the two lowest probability classes were binned together as neutral, the two highest probability classes were merged as damaging, and the middle class was considered under the two scenarios mentioned above. MutPred does not explicitly classify the variants. Therefore three ranked classes were assigned as follows: All the variants which scored below the threshold advised in the tool's documentation were considered as neutral. Variants scoring above the threshold for which MutPred additionally offered hypotheses about the nature of the mutation's effect comprised the damaging class. Variants scoring above the threshold, but lacking any hypothesis about the mutation's effect, were treated as the intermediate class, to be considered under the two scenarios.

4.2.4 Consensus classification development

In order to evaluate the potential benefit of incorporating outputs from multiple tools to improve predictions, two consensus approaches were implemented: a weighted majority vote score (WMV) and a Support Vector Machine (SVM).

Weighted Majority Vote

The WMV approach assigned a numerical value (V_i) to each of the three defined classes (damaging, intermediate and neutral) from each of the selected tools. The value +2 was assigned for the damaging class, +1 for the intermediate class,

and -2 for the neutral class. A score of 0 was assigned when a method did not generate a prediction. The reason why the intermediate class was assigned a positive value is that the tools performed better when the intermediate class was assumed to count as damaging (see 4.3.1).

The weighted vote score was calculated by adding up the individual values generated for each of the tools incorporated in the consensus model:

$$WMV = \sum_i V_i \tag{4.6}$$

The threshold value for the WMV is set to 0. Negative values lead to classification of the variant as neutral and positive values as damaging. If the votes add up to exactly 0, then classification is not possible by this method.

Support Vector Machine

Support Vector Machines are a form of supervised machine learning and are suitable for finding out the optimal weight for each of a vector of values, such as the output scores of a collection of independent tools.

The SVM-based consensus classifier incorporates the raw output scores generated by SIFT, PolyPhen2, SNPs&GO and Mutation Assessor. SVMlight [191] was used to build the SVM model and perform the classification using either a linear kernel or a radial based function (RBF) kernel with a g parameter of 0.0625 and the default C parameter. A grid search for optimal C and g parameters for the RBF kernel demonstrated a broad range of similarly well performing value combinations. The predictor was tested by means of 10-fold cross-validation: The sum of the *positive* and *negative* sets was split into pairs of non-overlapping subsets (evenly from *positive* and *negative*) and then one subset of each pair was used to train the model and the other to evaluate the accuracy. The training was performed in quadruplicate, changing the proportion of the data used to train the model ($2,000$, $5,000$, $10,000$ and $13,000$ variants). The results were very consistent across all 40 iterations with the standard deviation

< 0.01, and the accuracy improving less than 0.005 between the model trained on $2,000$ variants and the model trained on $13,000$ variants. The final classifier that has been made available to the academic public was trained on the entire dataset.

4.2.5 Development and Availability

Both the WMV and the SVM method are available under the name "CoVEC" (Consensus Variant Effect Classification) and have been implemented in the PERL scripting language. CoVEC is available both via a website interface[2] hosted on the departmental server, and as source code hosted on SourceForge[3] [192].

On the website interface for CoVEC, either the pre-obtained scores (for the SVM) or classifications (for the WMV) can be entered on the query form. The data is passed via an HTTP POST request to a CGI script, written in PERL. The WMV method is implemented directly in the CGI script, whereas scores intended for the SVM classifier are formatted and forwarded to the SVMlight executable, a copy of which is present locally on the server. The result is displayed in a simple web-page, both qualitatively (damaging or neutral) and quantitatively (WMV score or SVM score). The web interface is configured to accept a single variant per query.

The distributable source package for CoVEC is organised as a collection of PERL scripts and modules and is designed to allow batch queries and facilitate integration into custom software pipelines. Four PERL modules are implemented, one for each of SIFT, PolyPhen2, SNPs&GO and Mutation Assessor, which extract the data from each respective tool's individual batch output file. Two additional modules implement the automated query of the servers of SNPs&GO and Mutation Assessor, as these tools' websites do not support batch queries. The modules are flexible in terms of input data formatting, as

[2] http://www.dcs.kcl.ac.uk/pg/frousiok/variants/index.html
[3] http://sourceforge.net/projects/covec/files

a pre-emptive measure against updates to the individual methods' output formats. All parameters can be overridden by the user, but the most likely values for these parameters, at the time of the release, have been set by default. Each module comes with full PerlDoc documentation explaining its access methods and parameters. This library of modules is supplemented by a collection of three PERL scripts, which serve both as examples to developers on how to use the modules and as fully functional programs aimed at the end-user. Two of these scripts build on the respective two modules that send queries to SNPs&GO and Mutation Assessor, whereas the third script implements the extraction of results from the outputs of the individual methods, using the respective four modules. With regards to the actual consensus prediction, the WMV method is implemented as a standalone script in PERL, available in the distribution. SVMlight, on the other hand, is third party software, and is therefore not bundled in the CoVEC distribution. However, it is free to use academically and is available both as pre-compiled binaries and as source code. Users are instructed to download it separately. Both the linear and RBF models for SVMlight are included in the CoVEC distribution. Finally, the documentation of the CoVEC distribution is completed with a README manual and a collection of example data files.

4.3 Results and Discussion

4.3.1 Evaluation of the selected tools

The performance of the seven individual tools (SIFT, PolyPhen2, MutPred, SNPs&GO, PANTHER, PhD-SNP and Mutation Assessor) was assessed on a balanced dataset of $15,570$ variants, consisting of the *positive* and *negative* sets described in 4.2. Condel and CAROL are consensus tools and their performance results will be presented in section 4.3.2, along with the results of this work's newly proposed consensus strategies.

The proportion of variants classified by each of the seven individual methods

is > 0.9, except for PANTHER (Table 4.6). SNPs&GO, PhD-SNP and MutPred reported predictions for nearly all of the variants submitted. PANTHER's low proportion (0.68) may be caused by some variants not falling in positions covered by the multiple sequence alignments in its library [80]. Indeed, the authors of SNPs&GO, through which we obtained our PANTHER predictions, reported a similarly low proportion of classified variants (0.76) when benchmarking PANTHER [79]. MutPred demonstrated the highest correlation and the largest AUC, closely followed by SNPs&GO.

As explained in 4.2.3 two scenarios were considered in making the tools' predictions simulate binary classification. Under the first scenario, the intermediate class was considered as neutral and under the second it was considered as damaging. The results for these two scenarios, using the standard definitions of the performance metrics, is presented in Table 4.6.

All tools displayed the same or better performance in the second scenario, in which the intermediate class was considered as damaging, as demonstrated by the rise in accuracy. Therefore, from this point on, all subsequent measurements of performance will be under this scenario.

Considering the intermediate class as damaging, MutPred displayed the highest sensitivity of all of the tested classifiers (0.94) and the highest accuracy (0.92). However, SNPs&GO demonstrated the highest specificity (0.95) (Table 4.6).

With regards to the use of coordinates from the NCBI36/hg18 assembly or from the newer NCBI37/hg19 assembly, SIFT and PolyPhen2 (subscripts a and b in Table 4.6) displayed no change in overall accuracy. However, PolyPhen2 exhibited nearly identical performance with either assembly, whereas SIFT was able to classify a larger proportion of the variants when the coordinates were ported to the newer assembly, as opposed to using the original coordinates from the older assembly. SIFT also showed a small drop in sensitivity and rise in specificity with the newer assembly.

PolyPhen2's two versions of the classifier (HumVar and HumDiv, see 4.2.1,

		Scenario 1				Scenario 2			
Tools	*PC*	*Sens*	*Spec*	*Q2*	*MCC*	*Sens*	*Spec*	*Q2*	*MCC*
$SIFT_a$	0.87	0.63	0.91	0.71	0.58	0.78	0.80	0.79	0.58
$SIFT_b$	0.93	0.64	0.88	0.75	0.55	0.73	0.86	0.79	0.59
PNTH	0.68	0.69	0.84	0.75	0.52	0.69	0.84	0.75	0.52
PhD	1.00	0.62	0.78	0.70	0.41	0.62	0.78	0.70	0.41
S&G	1.00	0.71	0.95	0.83	0.68	0.71	0.95	0.83	0.68
MP	1.00	0.56	0.95	0.73	0.56	0.94	0.90	0.92	0.84
PPH_a	0.92	0.71	0.87	0.74	0.61	0.84	0.77	0.80	0.61
PPH_b	0.93	0.71	0.87	0.74	0.62	0.84	0.77	0.80	0.61
PPH_c	0.92	0.61	0.94	0.73	0.59	0.76	0.86	0.81	0.62
PPH_d	0.91	0.71	0.87	0.74	0.62	0.84	0.76	0.80	0.60
PPH_e	0.96	0.71	0.87	0.74	0.62	0.84	0.76	0.80	0.60
M/A	0.90	0.34	0.98	0.57	0.42	0.78	0.85	0.81	0.62

Table 4.6: Performance results for the seven individual tools. *PC*: Proportion of queries classified. Sensitivity (*Sens*), Specificity (*Spec*), Accuracy (*Q*2) and correlation (*MCC*) measured under the two scenarios. *Scenario 1*: The intermediate class was considered as neutral. *Scenario 2*: The intermediate class was considered as damaging. $SIFT_a$, PPH_a : SIFT and PolyPhen2, variants submitted as nucleotide substitutions with NCBI36/hg18 coordinates. $SIFT_b$, PPH_b : SIFT and PolyPhen2, variants submitted as nucleotide substitutions with NCBI37/hg19 coordinates. PPH_c: PolyPhen2, variants submitted as nucleotide substitutions with NCBI36/hg18 coordinates and using the HumVar predictor. PPH_d: PolyPhen2, variants submitted as amino acid substitutions. PPH_e: PolyPhen2, variants submitted as amino acid substitutions along with the corresponding amino acid sequences. For PolyPhen2, the HumDiv predictor was used, except where otherwise stated. *S&G*: SNPs&GO. *M/A*: Mutation Assessor. *PhD*: PhD-SNP. *PNTH*: PANTHER.

subscripts a and c in Table 4.6) displayed the same accuracy with each other, but, as expected, the more general HumVar version had higher sensitivity whereas the more strict HumDiv had higher specificity. PolyPhen2 was also executed with the variants supplied as amino acid substitutions (subscripts d and e), with no change in any of the performance and accuracy. There was however a significant improvement in the proportion of variants classified when the amino acid sequence was supplied, which suggests that the automatic retrieval of the sequences by PolyPhen2 may be prone to errors.

4.3.2 Evaluation of consensus strategies

The evaluation of these seven prediction tools demonstrates how each of the different approaches has different attributes. Therefore, methods of combining the outputs from these tools were evaluated, in order to improve the predictive performance. The results are presented in Table 4.7.

Consensus methods for variant effect classification have been published before. Condel [92] combines the output scores from PolyPhen2, SIFT and Mutation Assessor. Evaluation of this approach in the present study demonstrated that, in comparison to the three methods it combines, it performed better than SIFT and comparably to Mutation Assessor. It also showed a better accuracy and correlation than PolyPhen2, but the latter retained considerably higher sensitivity. Another consensus tool, CAROL [84], which combines SIFT and PolyPhen2, showed similar performance to Condel.

For the development of a new consensus approach, combinations of different subgroups of the six individual classifiers were evaluated using either the Weighted Majority Vote method (WMV) or a support vector machine (SVM) approach. MutPred was excluded, despite appearing to be the best tool out of the 7, because of the direct overlap between its training data and this study's HGMD-derived *positive* dataset. To avoid the bias of testing the SVM classifier on the data it was trained on, the values listed in Table 4.7 are averaged from 40-fold cross-validation (see 4.2.3).

Tools	*PC*	*Sens*	*Spec*	*Q2*	*MCC*
Condel	1.00	0.77	0.88	0.83	0.66
CAROL	0.99	0.79	0.85	0.82	0.64
WMV:					
$SIFT_a$, PPH_a	0.89	0.81	0.85	0.83	0.66
$SIFT_b$, PPH_b	0.88	0.79	0.88	0.84	0.68
PPH_e, S&G, PNTH, PhD, M/A	0.97	0.75	0.93	0.84	0.69
PPH_e, S&G, PhD, M/A	0.97	0.74	0.94	0.84	0.70
PPH_e, S&G, M/A	0.96	0.80	0.92	0.86	0.73
$SIFT_b$, PPH_e, S&G, PNTH, PhD, M/A	0.98	0.74	0.93	0.84	0.69
$SIFT_b$, PPH_e, S&G, M/A	0.97	0.76	0.92	0.84	0.69
$SIFT_b$, PPH_e, M/A (like Condel)	0.96	0.79	0.89	0.84	0.67
$SIFT_b$, PPH_e (like CAROL)	0.89	0.79	0.88	0.84	0.68
SVM (cross-validation average):					
linear	1.00	0.83	0.90	0.87	0.74
RBF	1.00	0.84	0.89	0.87	0.74

Table 4.7: Performance results for the consensus methods. Proportion of queries classified PC, Sensitivity $Sens$, Specificity $Spec$, Accuracy $Q2$ and correlation MCC measured with the intermediate class considered as damaging. $SIFT_a$, PPH_a : SIFT and PolyPhen2, variants submitted as nucleotide substitutions with NCBI36/hg18 coordinates. PPH_e: PolyPhen2, variants submitted as amino acid substitutions along with the corresponding amino acid sequences. For PolyPhen2, the HumDiv predictor was used. *S&G*: SNPs&GO. *M/A*: Mutation Assessor. *PhD*: PhD-SNP. *PNTH*: PATHER.

Using the WMV, combinations of methods that used amino acid substitutions as input generally fared better than those that used nucleotide substitutions. The performance also improved as methods that performed weakly individually (Table 4.6) were omitted from the consensus. The highest WMV accuracy and correlation was obtained from the combination of PolyPhen2, SNPs&GO and Mutation Assessor (Table 4.7). The specificity was high and comparable to that of SNPs&GO, which is the most specific of all the tools benchmarked in this study, whereas the sensitivity was higher than two of the constituent methods, SNPs&GO and Mutation Assessor, but not as high as PolyPhen2.

In order to enable a direct comparison of the WMV method with the weighted score methods employed by Condel and CAROL, the WMV combinations of SIFT, PolyPhen2 and Mutation Assessor (as in Condel), and SIFT and PolyPhen2 (as in CAROL), were among the WMV combinations evaluated. Table 4.7 clearly shows that the WMV performed equally well to Condel and CAROL. This is particularly interesting, considering the fact that both Condel and CAROL use complex fine-weighting schemes for the constituent tools' scores, whereas the WMV method is, by comparison, very simplistic. This indicates that the original confidence labels provided by the individual tools are sufficient or, conversely, that the alternative weighting systems implemented in Condel and CAROL offer no significant improvement with regards to insight into the reliability of the individual predictions. The downside of this simplistic approach, is that the WMV score can take only a small and finite number of possible values. When this score equals the threshold, the variant cannot be classified; a problem that sets back the WMV method in terms of the proportion of classified variants, compared to Condel and CAROL.

For the SVM-based classifier, the same tools were used as for the best WMV combination, (PolyPhen2, SNPs&GO and Mutation Assessor) with the addition of SIFT. MutPred was excluded for the same reason as above, while PhD-SNP and PANTHER were excluded after observing their poor contribution to the

Tools	*Sens*	*% FN*	*% Unknown*
SIFT	0.74	0.22	0.05
PANTHER	0.60	0.23	0.17
PhD-SNP	0.70	0.30	0.00
SNPs&GO	0.85	0.15	0.00
MutPred	0.95	0.05	0.00
PolyPhen2	0.88	0.12	0.00
M. Assessor	0.64	0.19	0.17
Condel	0.81	0.19	0.00
CAROL	0.85	0.15	0.00
WMV (PPH,S&G,M/A)	0.84	0.13	0.03
SVM (linear)	0.91	0.09	0.00

Table 4.8: Classification results on a validation set composed of 4985 disease-associated amino acid substitutions from PhenCode. Sensitivity ($Sens$, percentage of True Positives), percentage of False Negatives ($\%FN$) and percentage of unclassified variants. For WMV, the previously determined best-scoring combination was used (see Table 4.7)
.

WMV consensus. The resulting SVM model provided elevated accuracy and correlation in comparison to the most accurate WMV combination and outperformed both Condel and CAROL. The use of the linear and RBF kernels produced almost identical predictions. In comparison to the individual tools, the SVM approach matched the highest observed sensitivity (PolyPhen2) and provided better accuracy and correlation (except for MutPred). It also provided the second highest observed specificity (alongside MutPred) out of all the tools discussed in this article, coming second to SNPs&GO.

4.3.3 Additional validation

As experimentally validated variants are relatively few, there is a high risk of the datasets used here overlapping the training datasets of the tested methods and causing biases, especially with regards to disease-causing variants. Therefore, an additional separate set of 4985 disease-associated variants was collected (see 4.2.2) and submitted for individual and consensus classification. The results are shown in Table 4.8.

SIFT and MutPred demonstrated sensitivity very similar to the one mea-

sured on the HGMD-derived dataset (Table 4.6). This is particularly notable, considering the initial concern that the overlap between MutPred's training set and the benchmark dataset used in this study would bias the results in its favour. PolyPhen2 and WMV showed a small increase in sensitivity, whereas PhD-SNP, SNPs&GO and the SVM classifier showed a more drastic increase. Mutation Assessor and PANTHER were the only ones for which sensitivity was lower than measured on the HGMD-derived dataset. Despite the changes, the overall hierarchy was not fundamentally affected; MutPred maintained its lead, followed by the SVM classifier.

4.4 Conclusions and perspectives

Current DNA and RNA sequencing technologies make knowledge of the sequence of a nucleic acid broadly affordable and accessible. Genetic studies of inherited diseases and cancer make use of this technology in order to identify the exact genetic causes of these conditions and understand the mechanisms involved, with the aim to provide medical treatments and cures. As a result from these sequencing projects, a large number of novel variants arises, not all of which are relevant to the condition being studied. The number of variants is prohibitive for experimental validation, creating the need for bioinformatics solutions with which to prioritise the candidate culprits and focus the experimental efforts. A number of tools addressing this task have been developed. A selection of them, consisting of SIFT, PANTHER, PhD-SNP, SNPs&GO, MutPred, PolyPhen2 and Mutation Assessor, was benchmarked in this study, for the purpose of building a consensus classifier. Two consensus classifiers have been proposed and compared to the existing consensus tools Condel and CAROL.

The first proposed classifier is based on a simple voting system, whereby classification of a variant as either disease-causing or neutral is based on majority. Confident classifications by the combined tools are assigned two votes each, whereas less confident classifications are assigned one vote. This system

has been shown to perform equally well to the two existing consensus methods, Condel and CAROL, both of which employ sophisticated weighting schemes for the raw score outputs of the tools they combine.

The second proposed classifier is based on machine learning and employs a Support Vector Machine trained on monogenic disease–causing variants from the Human Gene Mutation Database and putatively neutral variants extracted from the 1000 Genomes Project pilot data. This classifier has been shown to outperform all the consensus tools and offer a more even balance of sensitivity and specificity compared to individual tools, as well as higher overall accuracy and correlation than all but one of the benchmarked tools.

The use of bioinformatics tools to evaluate the functional consequences of newly discovered variants is a core step in the study of the underlying genetic mechanisms of inheritable diseases. Although monogenic diseases are limited in number, relatively simple in analysis and likely to be all solved in the near future, there is a large selection of complex diseases that need to be understood. These are commonly approached with genome–wide association studies (GWAS), which require large positive and control groups in order to achieve statistical significance, which are hard to obtain for rare diseases. Additionally, it is becoming increasingly difficult to define control datasets, as the variation databases grow faster than variants can be fully studied and understood, creating a high risk of false controls. Use of variant evaluation tools may be a more viable alternative strategy, as our knowledge base of gene function and interactions grows.

The current tools, including the proposed and reviewed consensus strategies, stop at classifying variants as either damaging or neutral, with varying definitions of what either term means. More helpful would be the ability to quantify the consequences of variants and predict the type and range of their effects. Some methods, like MutPred, have already moved in that direction and offer the most accurate and informative results out of the available tools. Eventually, such programs should be able to retrieve all the functions and interactions of

genes at all levels and assess and visualize the effects of variants, enabling scientists to identify even weak associations with various phenotypes without the need of large study groups. Standing in the way of achieving this, more than a lack of methodologies, is our still limited knowledge of the complex interaction networks at the molecular level, as well as the fragmentation of this knowledge across many resources with varying format consistency, varying levels of completeness and varying standards of validation and update.

Availability

CoVEC (Consensus Variant Effect Classification) [192] is available to the academic public via a website[4] currently hosted on the CALCIUM.DCS.KCL.AC.UK departmental server and as source code on SourceForge[5].

[4]http://www.dcs.kcl.ac.uk/pg/frousiok/variants/index.html
[5]http://sourceforge.net/projects/covec/files

Chapter 5

Conclusions

5.1 Contributions of this thesis

From the obtainment of the raw sequencing data to the extraction of a meaningful interpretation of the results, several stages of analysis take place, each presenting its own computational challenges. The alignment of the sheer number of short reads created by the NGS platforms requires special algorithms and considerable processing power, while the presence of repeats in the genome complicates the unambiguous alignment of the reads to the reference genome and renders virtually impossible the *de novo* assembly of the reads into long contiguous sequences. The presence of sequencing and other errors further complicates both the alignment and the assembly. The resulting alignments or assembly must then be evaluated and quantified and any previously unknown variants must be detected. The presence of errors, the diploid nature of the genome and potentially the presence of contaminants contribute to the difficulty of this stage. Finally, the detected variants must be evaluated with regards to their potential function, but the number of previously unknown variants that are detected in each re-sequencing project is such, that experimental characterisation of every variant is not feasible and requires the use of computational tools.

This thesis contributes improvements to two of these analysis stages. For

the read mapping stage, an alternative scoring scheme is proposed, that uses the actual substitution frequencies observed among humans in order to build a substitution score matrix to help disambiguate alignments that otherwise appear equivalent under the match/mismatch model. This scoring scheme is implemented in the in-house algorithm *REAL* and demonstrated to have increased sensitivity over the match/mismatch model, particularly for shorter reads. This demonstrates that, despite the common practice of simply counting mismatches, there is sufficient reason to instead use scoring schemes that model the evolutionary relationship between the sequences, even for highly related short sequences, and that this can be done without an icrease in algorithm complexity. This conclusion echoes similar results from an independent study, which, however, did not include the *GC* mutability bias, and extends the benefits of the use of evolutionary scores from cross-species alignments to also cover same–species alignments. Although shorter reads are being progressively phased out, thanks to advances in the chemistry and electronics of current NGS methods and the appearance of the third generation of sequencing methods, this result will remain relevant as long as NGS platforms are still in use. More importantly, as demonstrated in this work, the use of a substitution score matrix enables the alignment of reads to more loci than possible with the mismatch count model, even though fairly lenient mismatch thresholds were used. The ability to detect activity at more loci while keeping the trade-off low could make a great difference in the search for rare or unknown variants.

For the variant evaluation stage, a series of widely used tools was benchmarked using a set of monogenic disease-causing variants and a set of probably neutral variants collected from commercial and public databases. Two consensus methods were subsequently proposed and demonstrated to offer improved classification accuracy, compared to most of the major individual tools considered in this work, as well as to both of the rival consensus tools, both of which were released around the same time as this work. Unlike the read alignment substitution matrix, that is tied to a technology, the evaluation of variants is not

tied to any particular sequencing technology and will remain a valuable asset in the research of the genetic causes of inherited diseases.

Additionally, *REAL* was used in a proof-of-concept work that used short RNAseq data to study genome-wide expression levels. As a result, the first isochoric transcriptome map was drawn, for three discrete adult mouse tissues, as well as three developmental stages of the same tissue. The map demonstrates areas of clear differences in the transcriptional activity between tissues and between developmental stages, where previously such information was available only for narrow scopes, as a result of the limitations of the experimental procedures available at the time. Additionally, this work conclusively dismisses the doubts that have been persistently expressed with regards to the correlation between local expression levels and the base composition of the genome. Resolving the debate over the mechanism of evolution of isochores was beyond the scope of this work. It is however my opinion that the two theories are not necessarily mutually exclusive, as one revolves heavily on a possible evolutionary mechanism whereas the other revolves mainly about a possible advantage for the local increase of *GC* and both are supported by convincing evidence. In all cases, isochores are a real compartmentalization of the genome, rather than an artifact of methodologies used, and the local base composition certainly can influence the transcriptional activity on a large scale.

5.2 Possible extensions and future research

As previously discussed, this work completely ignored the presence of gaps in real–life alignments, as *REAL* lacked a strategy for gapped alignment at the time. However gaps are not an uncommon event and are, in fact, very common in alignment of cDNA reads from RNAseq experiments, as a result of RNA splicing. An algorithm has since been developed [65] that partially handles the presence of gaps, but it has not been integrated into *REAL* and its accuracy has not been tested. Additionally, indels in exome sequencing and RNAseq often

appear with lengths that are multiples of 3, as a result of the triplet nature of the genetic code. It would therefore be interesting to test how a gap penalty scheme that favours gap lengths that are multiples of 3 performs, compared to the usual model of having a penalty for opening a gap and a constant penalty for extending a gap.

With regards to variant effect classification, both proposed consensus models are easily extensible so as to include other classification tools. Although many tools are currently available, leveraging annotation from different combinations of multiple resources, new tools and improved versions of the current classifiers may be released in the future. In this case, the chosen tools would need to be reconsidered and re-evaluated along with the newer ones. The vote-based method is readily extensible as it is, whereas the SVM-based method would require a new model to be trained afresh for a new combination of tools. The web interface is currently quite basic; it supports only one variant per query and the predictions need to be obtained manually from the individual tools, as it was designed only as a temporary implementation of the consensus methods. A fully fledged implementation would have to be migrated to a dedicated server location and could include features such as automatic integrated submission of a variant to the individual tools for prediction and automatic retrieval of the results or support of batch queries.

Finally, the question regarding the origins and role of isochores remains largely unanswered. As discussed, isochore compartmentalization using the base composition correlates with a large number of other features, so it is possible that the composition is not the primary characteristic of isochores but, instead, is one of the many results of one of the other known or even yet unknown features. A great amount of research remains to be done in order to understand how the different features of isochores interact with one another, before being able to understand how these interactions might have shaped the evolution of isochores. In the shorter term, the confidence that isochores do correlate with expression is highly relevant to the study of epigenomic influences on gene

regulation.

Appendices

Appendix A

Transcriptome map of mouse isochores

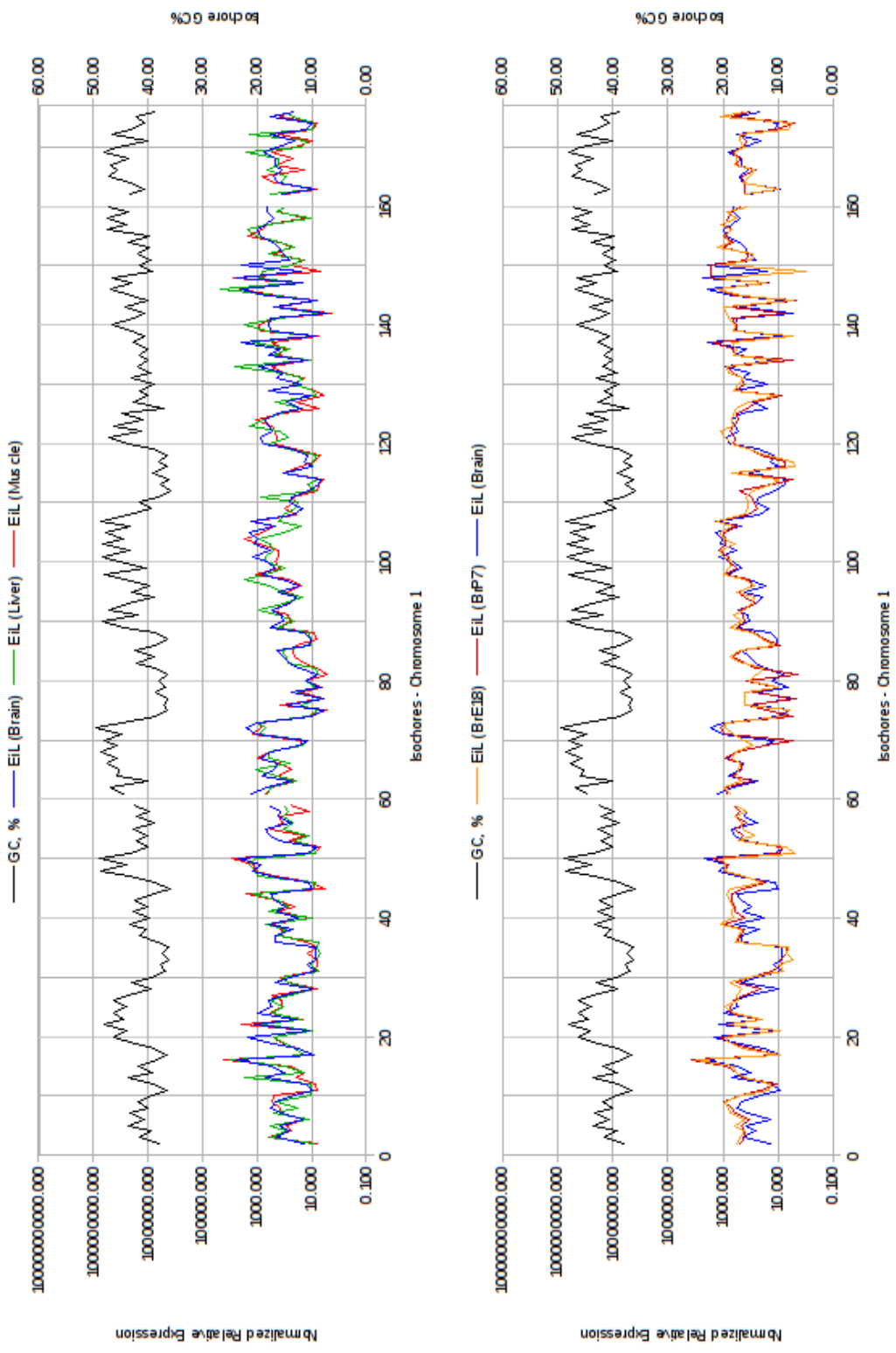


Figure A.1: *GC* content profile and expression profile along mouse chromosome 1, separately for each adult tissue and each developmental stage of the brain.

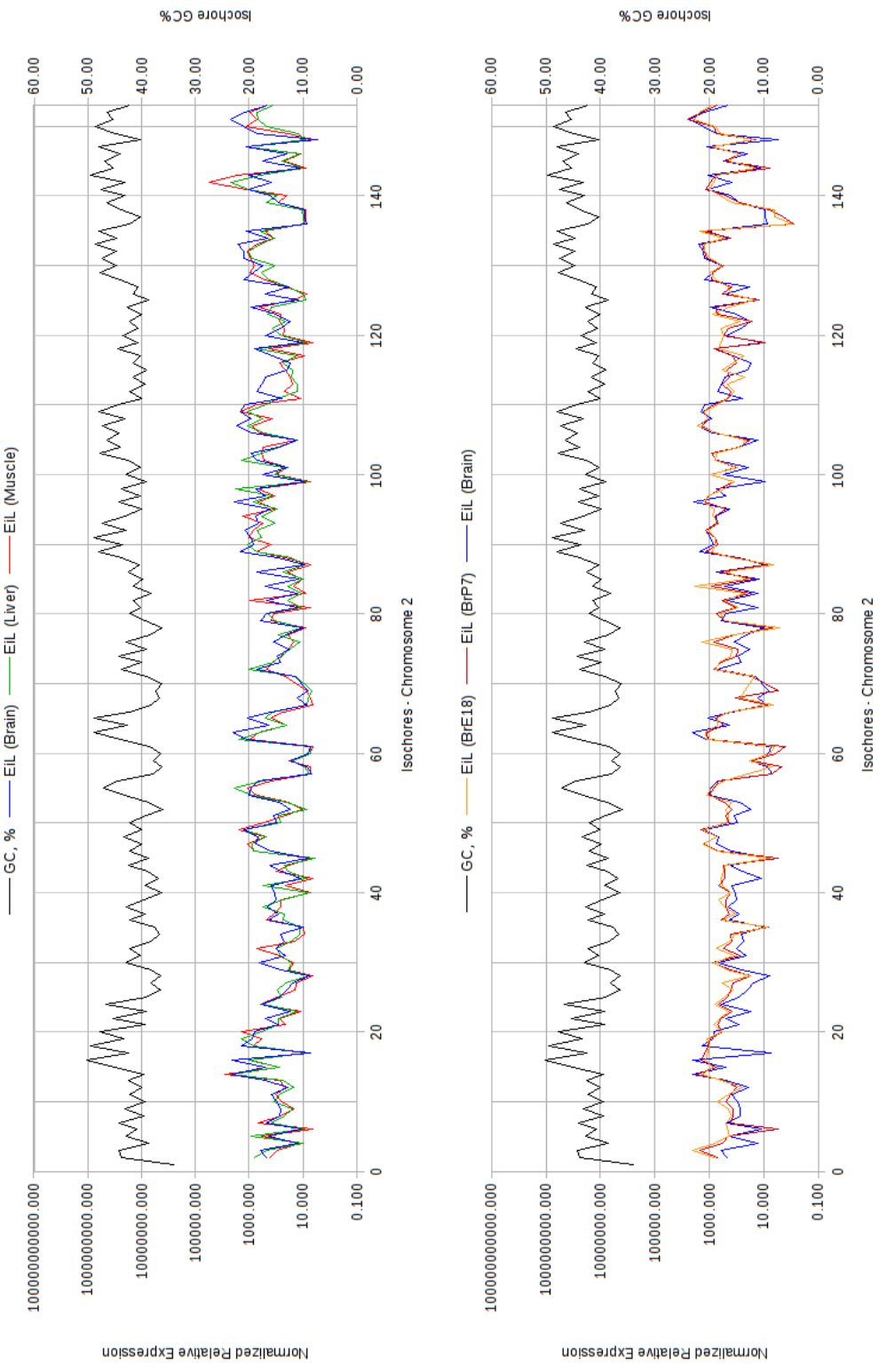


Figure A.2: *GC* content profile and expression profile along mouse chromosome 2, separately for each adult tissue and each developmental stage of the brain.

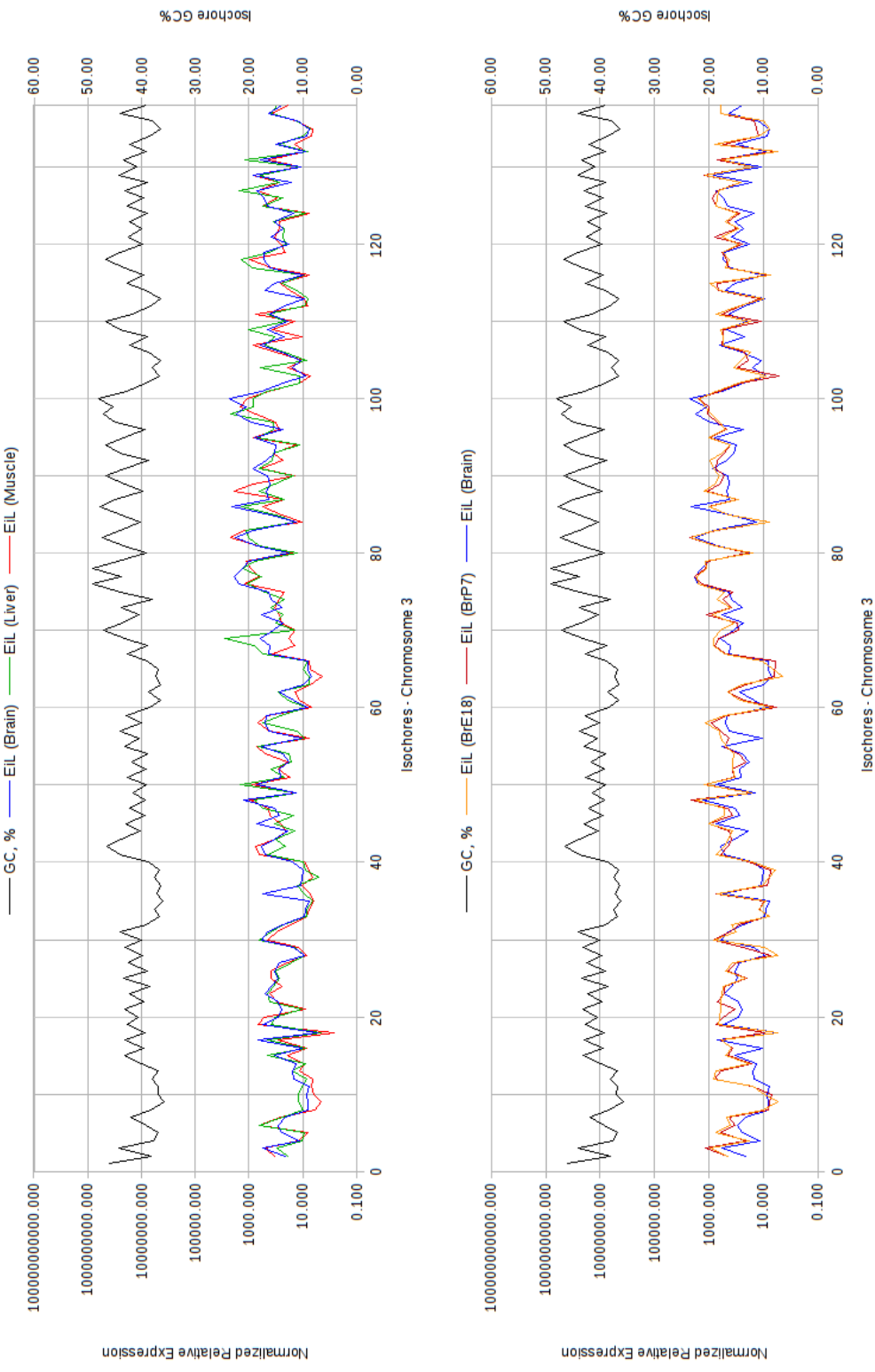


Figure A.3: *GC* content profile and expression profile along mouse chromosome 3, separately for each adult tissue and each developmental stage of the brain.

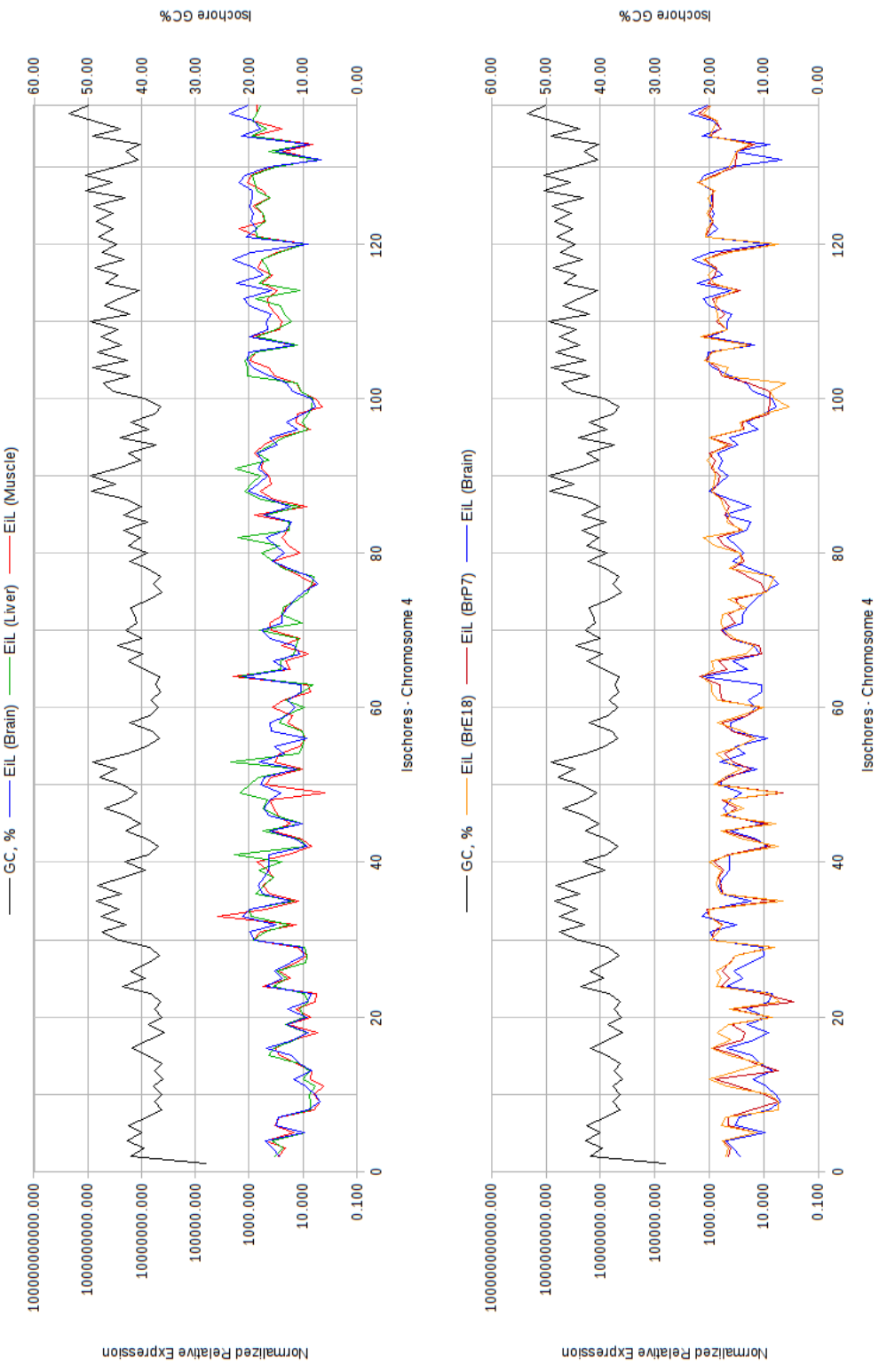


Figure A.4: *GC* content profile and expression profile along mouse chromosome 4, separately for each adult tissue and each developmental stage of the brain.

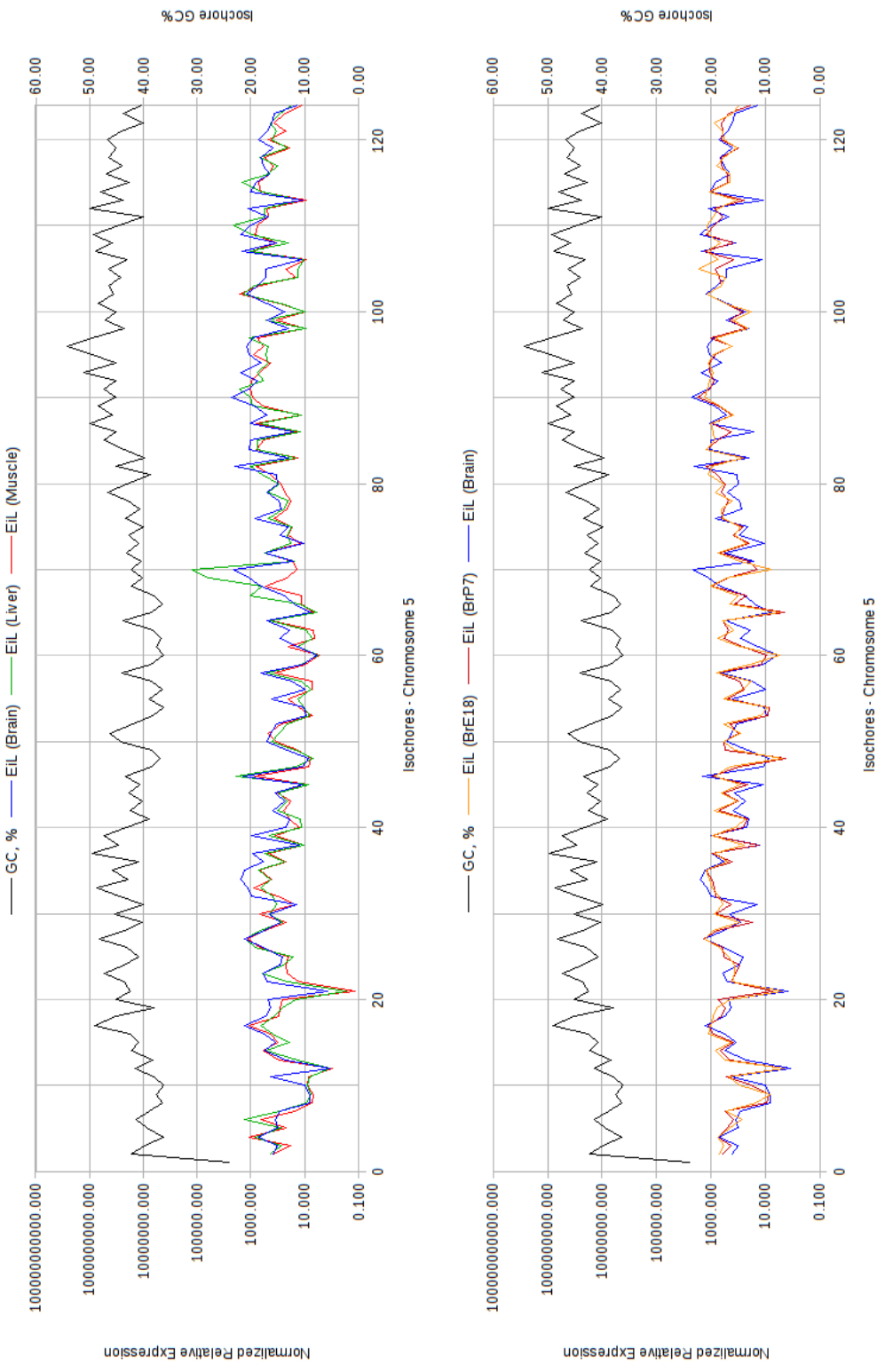


Figure A.5: *GC* content profile and expression profile along mouse chromosome 5, separately for each adult tissue and each developmental stage of the brain.

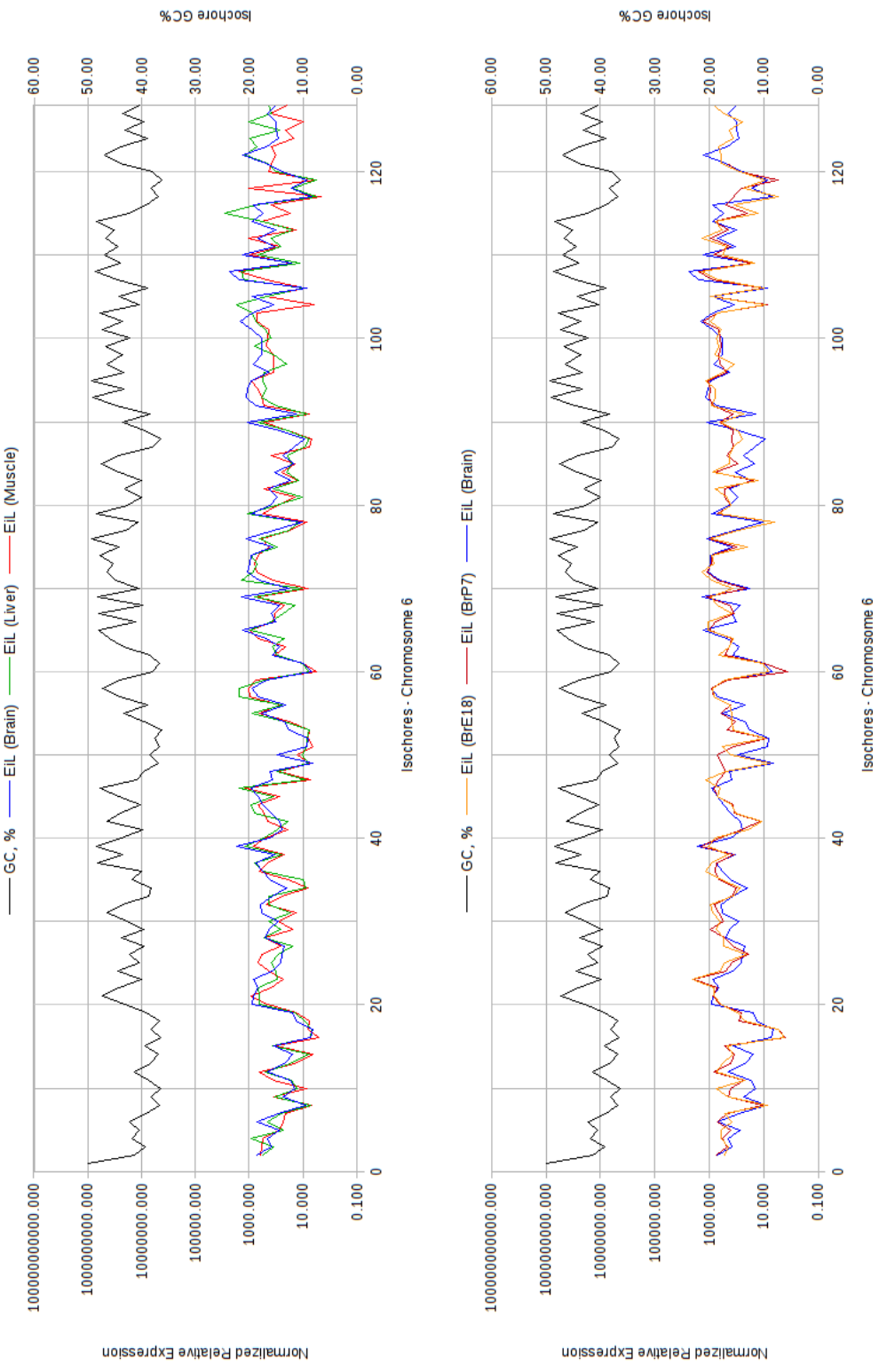


Figure A.6: *GC* content profile and expression profile along mouse chromosome 6, separately for each adult tissue and each developmental stage of the brain.

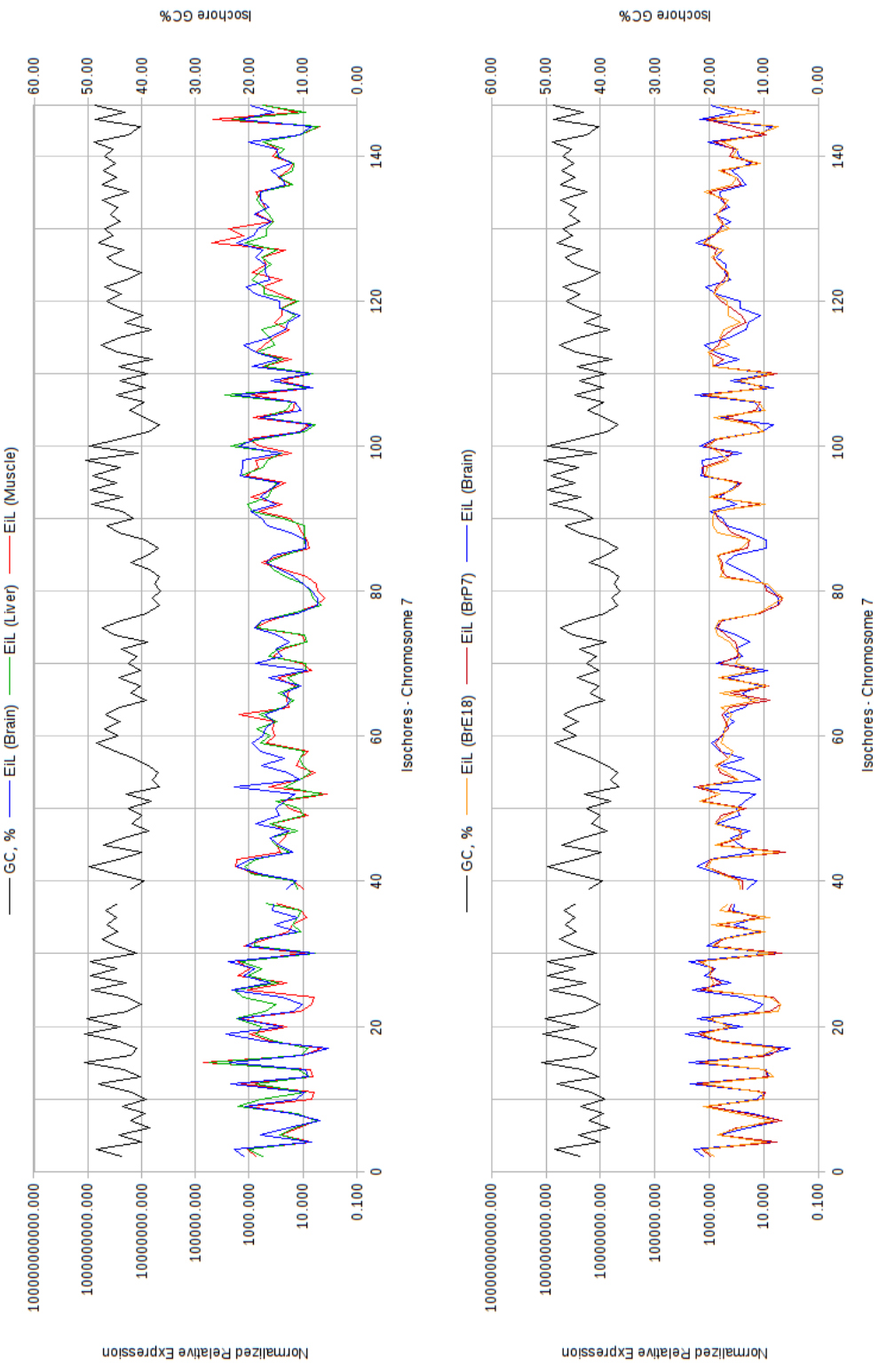


Figure A.7: *GC* content profile and expression profile along mouse chromosome 7, separately for each adult tissue and each developmental stage of the brain.

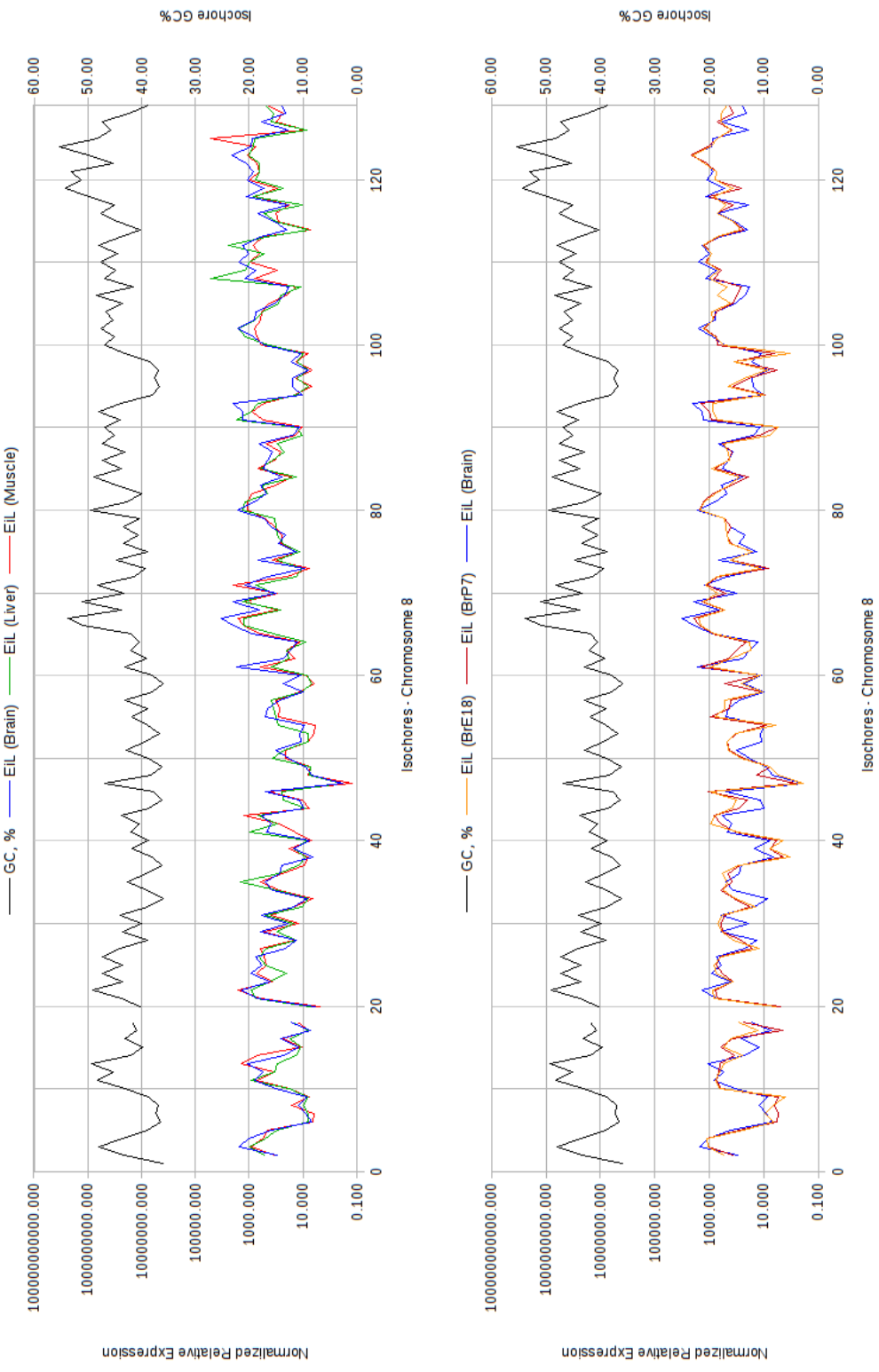


Figure A.8: *GC* content profile and expression profile along mouse chromosome 8, separately for each adult tissue and each developmental stage of the brain.

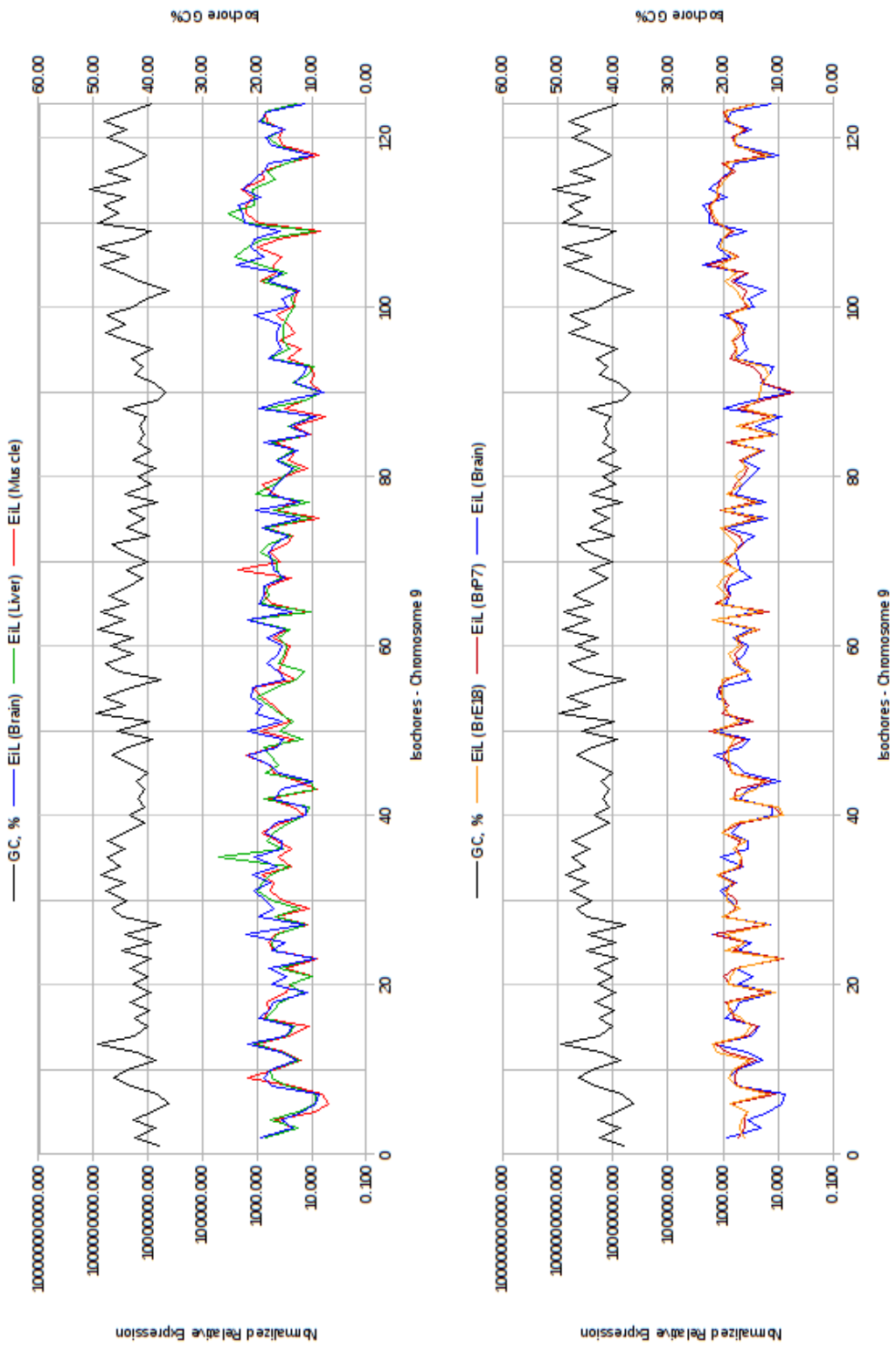


Figure A.9: *GC* content profile and expression profile along mouse chromosome 9, separately for each adult tissue and each developmental stage of the brain.

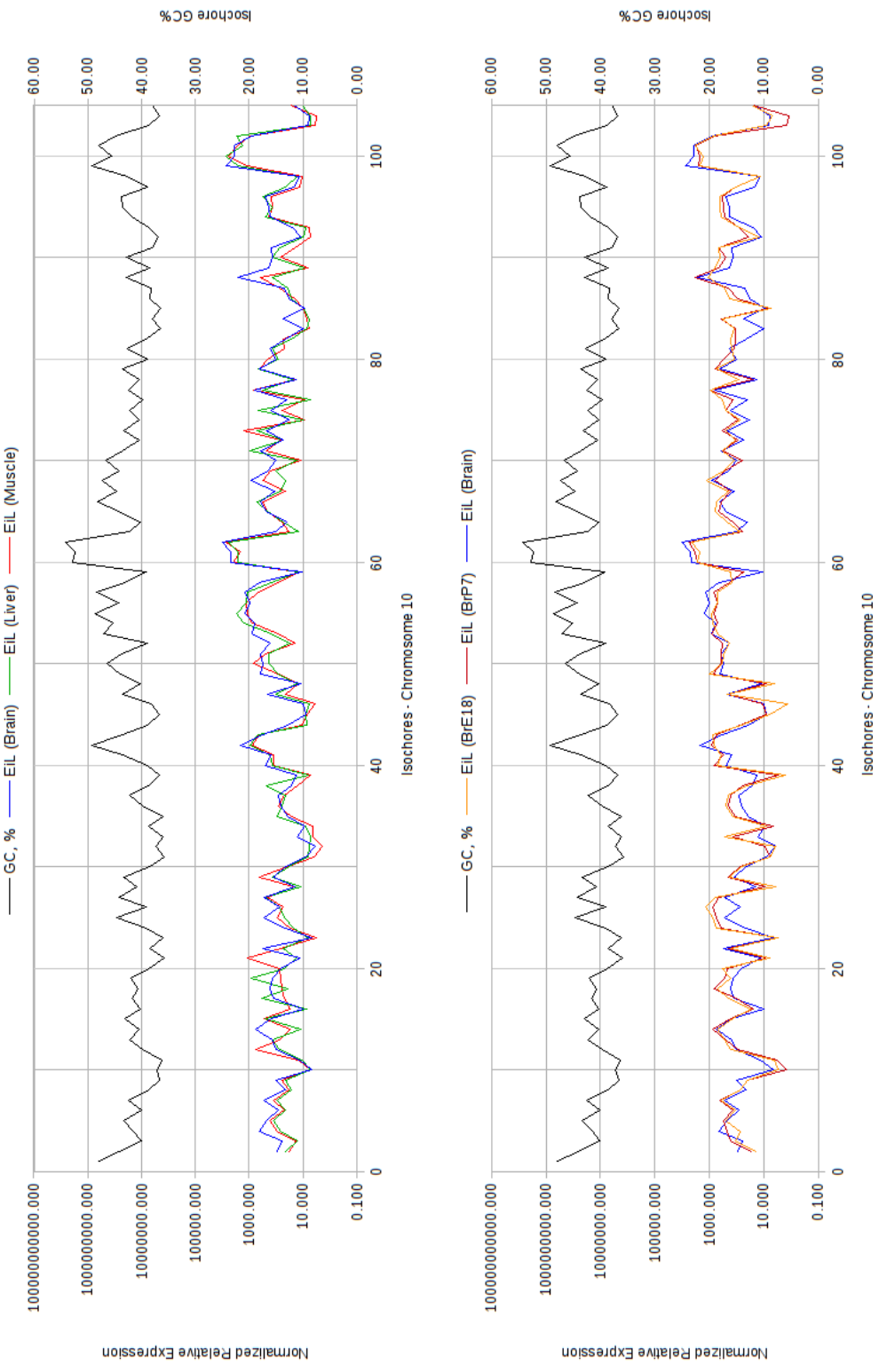


Figure A.10: *GC* content profile and expression profile along mouse chromosome 10, separately for each adult tissue and each developmental stage of the brain.

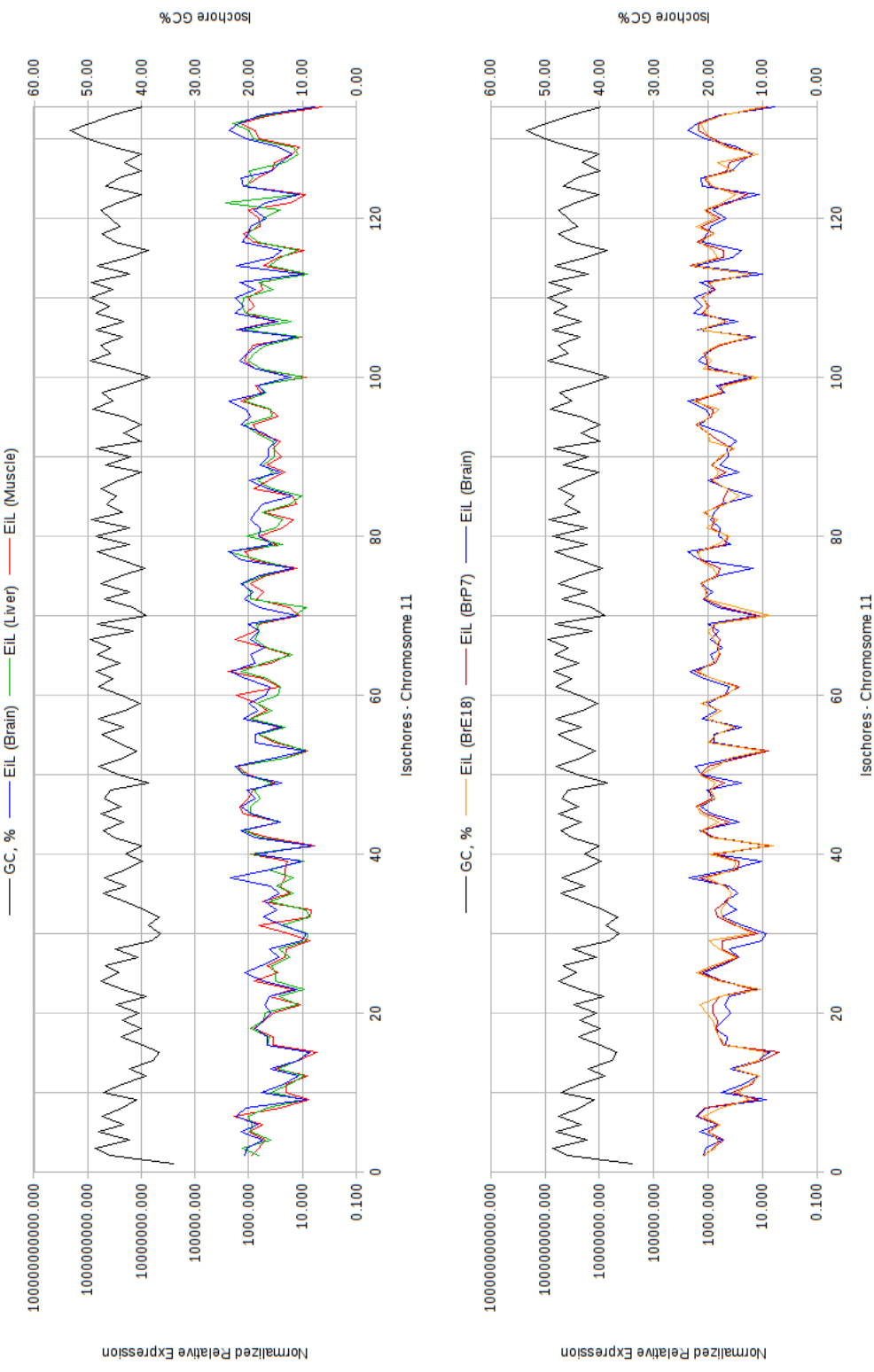


Figure A.11: *GC* content profile and expression profile along mouse chromosome 11, separately for each adult tissue and each developmental stage of the brain.

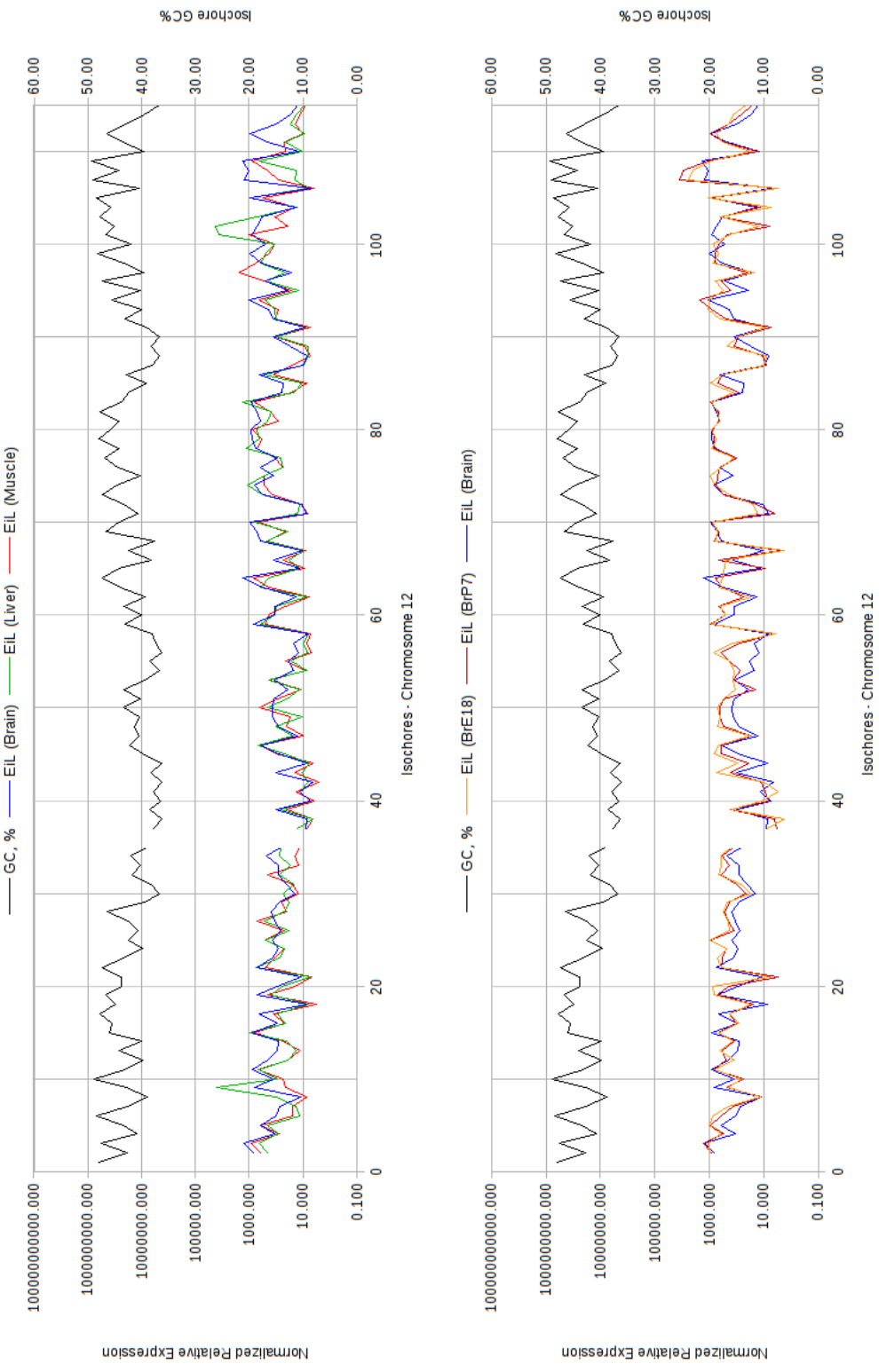


Figure A.12: *GC* content profile and expression profile along mouse chromosome 12, separately for each adult tissue and each developmental stage of the brain.

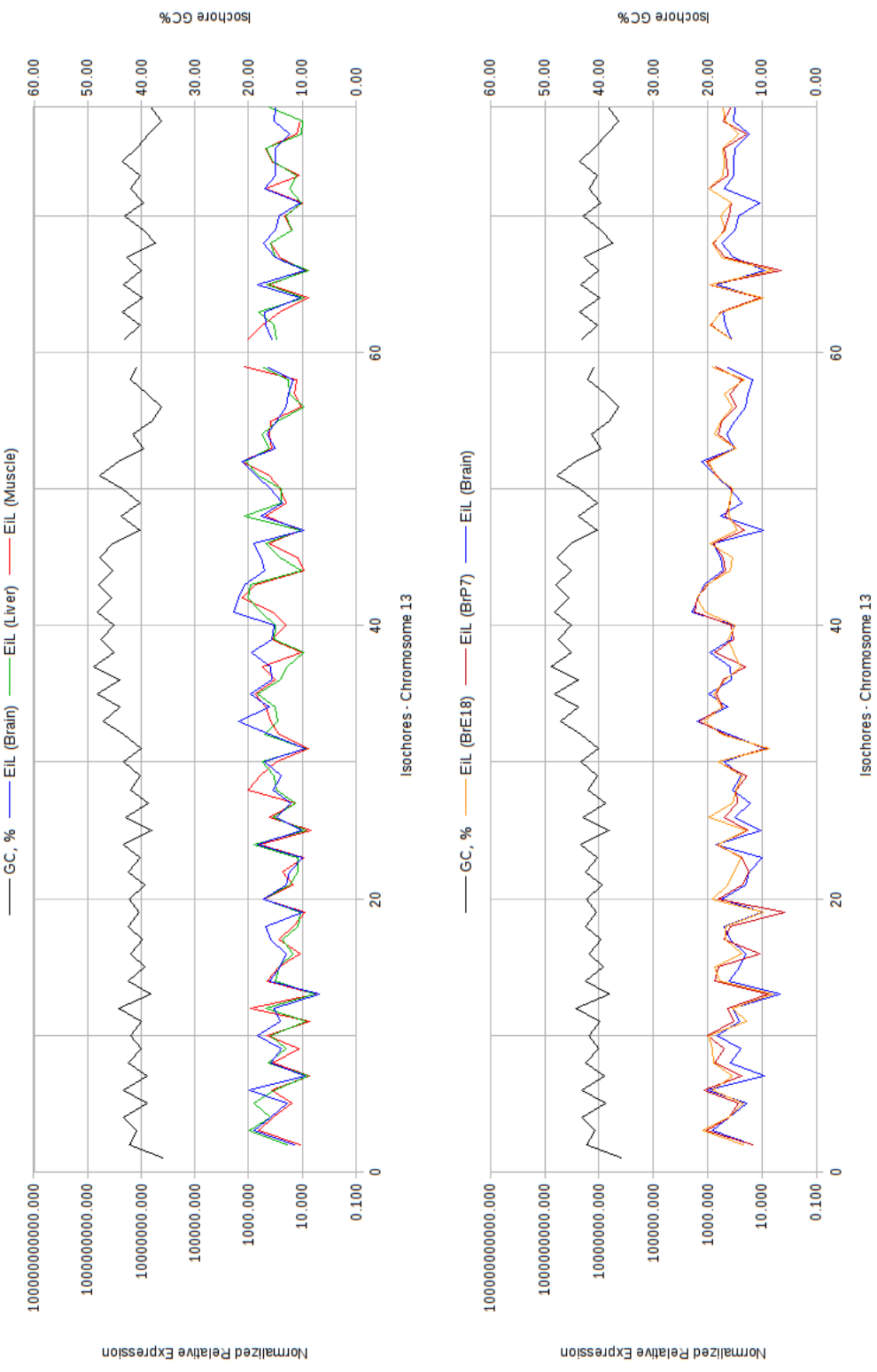


Figure A.13: *GC* content profile and expression profile along mouse chromosome 13, separately for each adult tissue and each developmental stage of the brain.

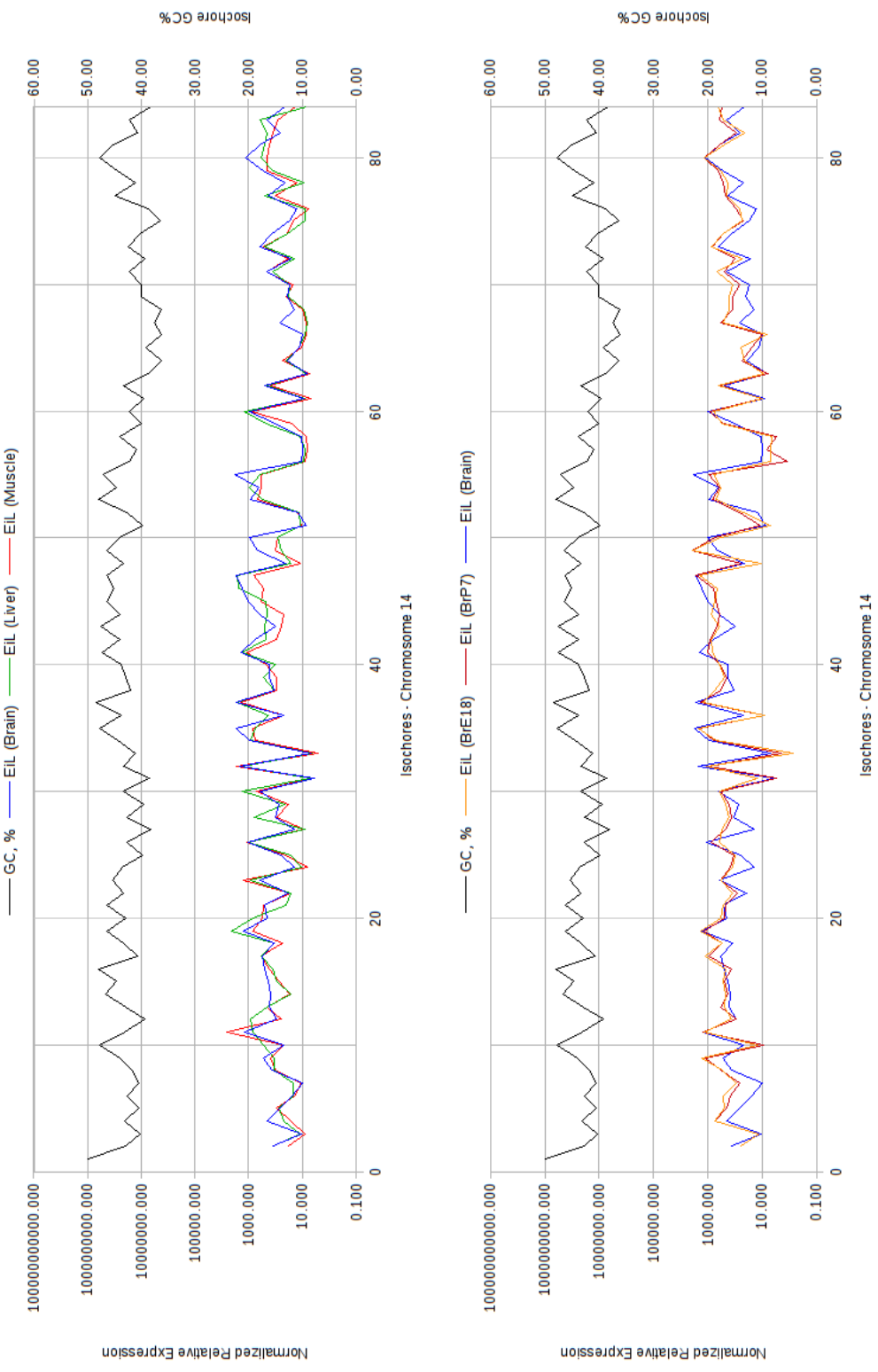


Figure A.14: *GC* content profile and expression profile along mouse chromosome 14, separately for each adult tissue and each developmental stage of the brain.

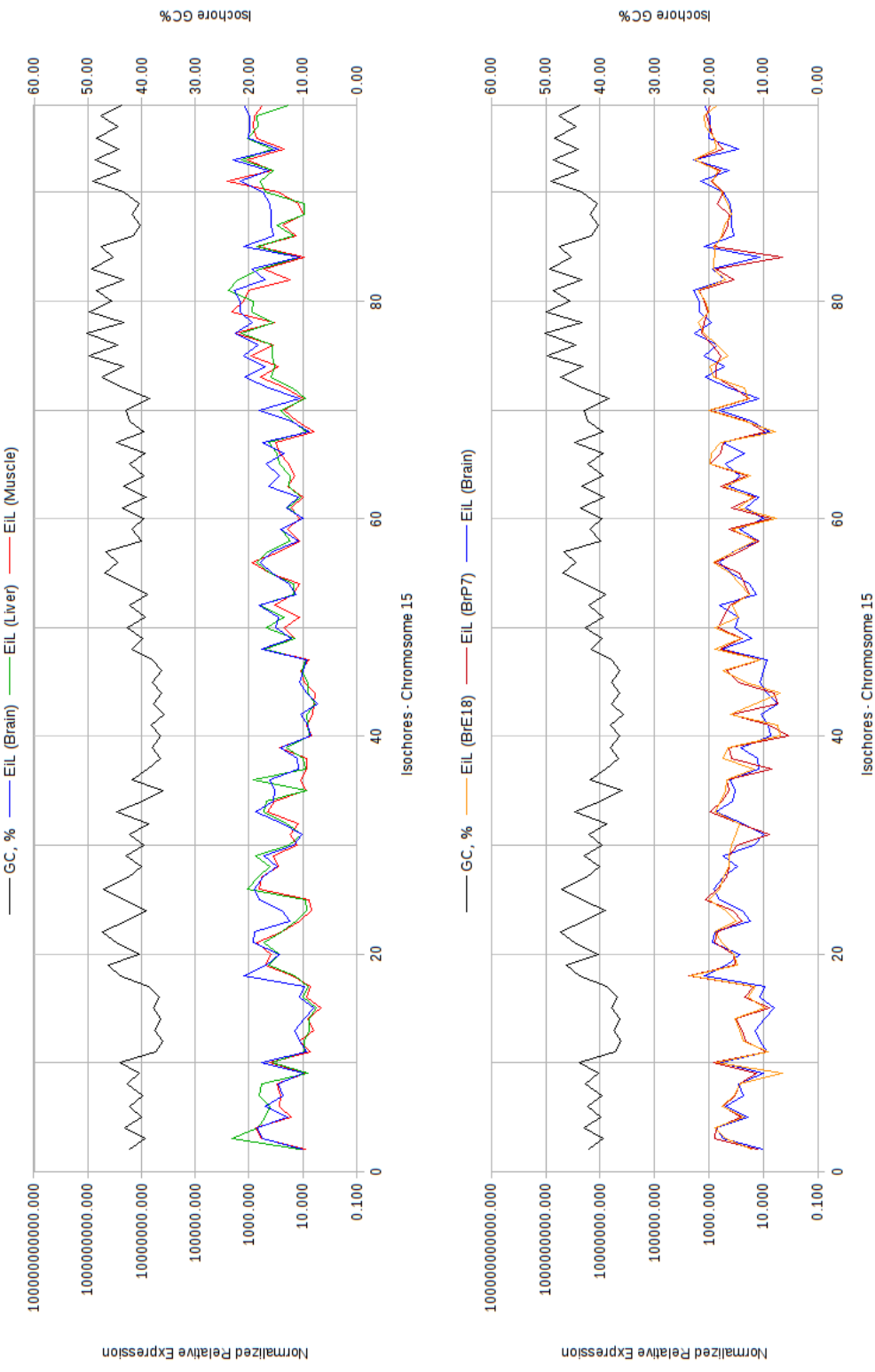


Figure A.15: *GC* content profile and expression profile along mouse chromosome 15, separately for each adult tissue and each developmental stage of the brain.

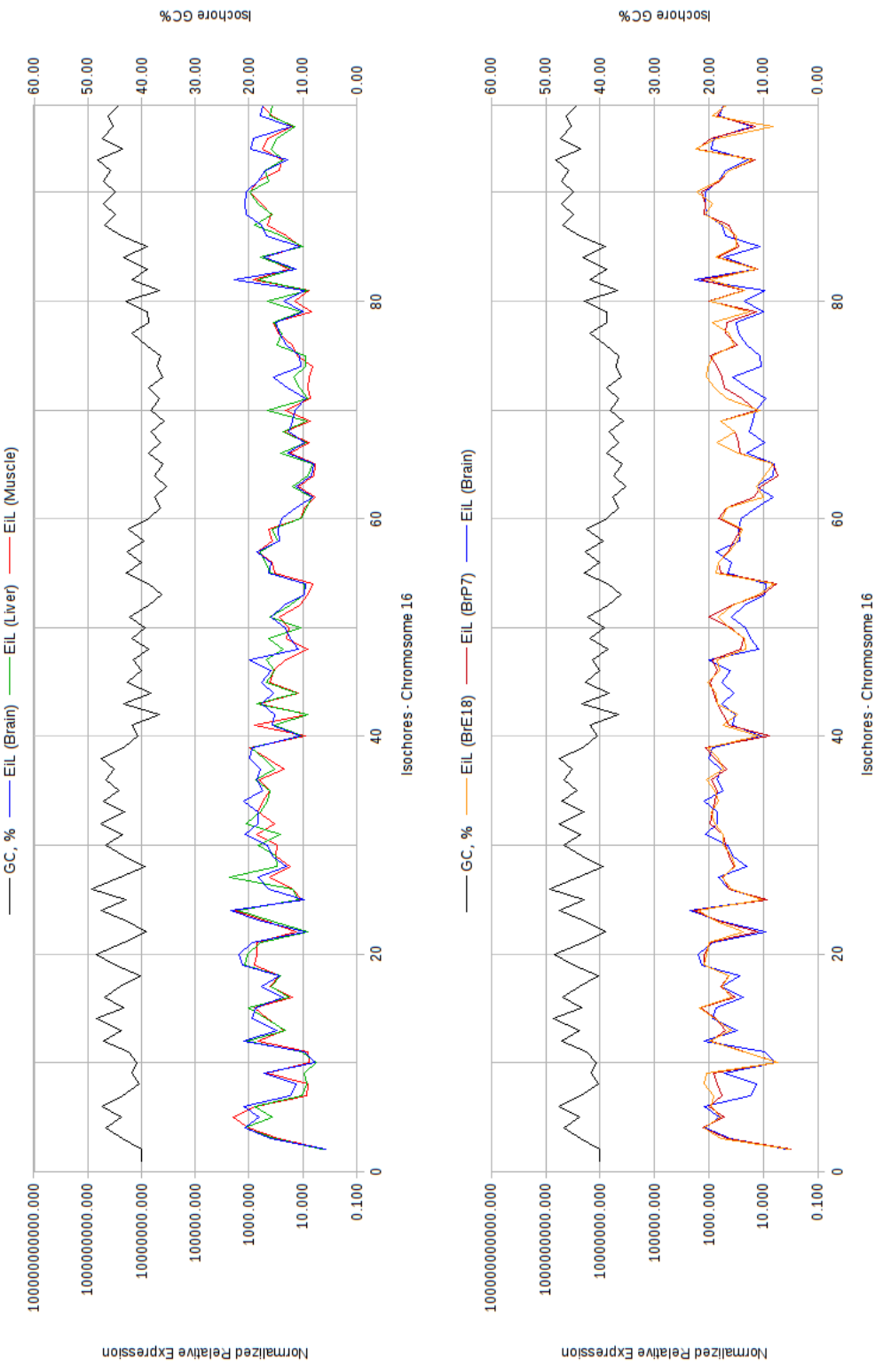


Figure A.16: *GC* content profile and expression profile along mouse chromosome 16, separately for each adult tissue and each developmental stage of the brain.

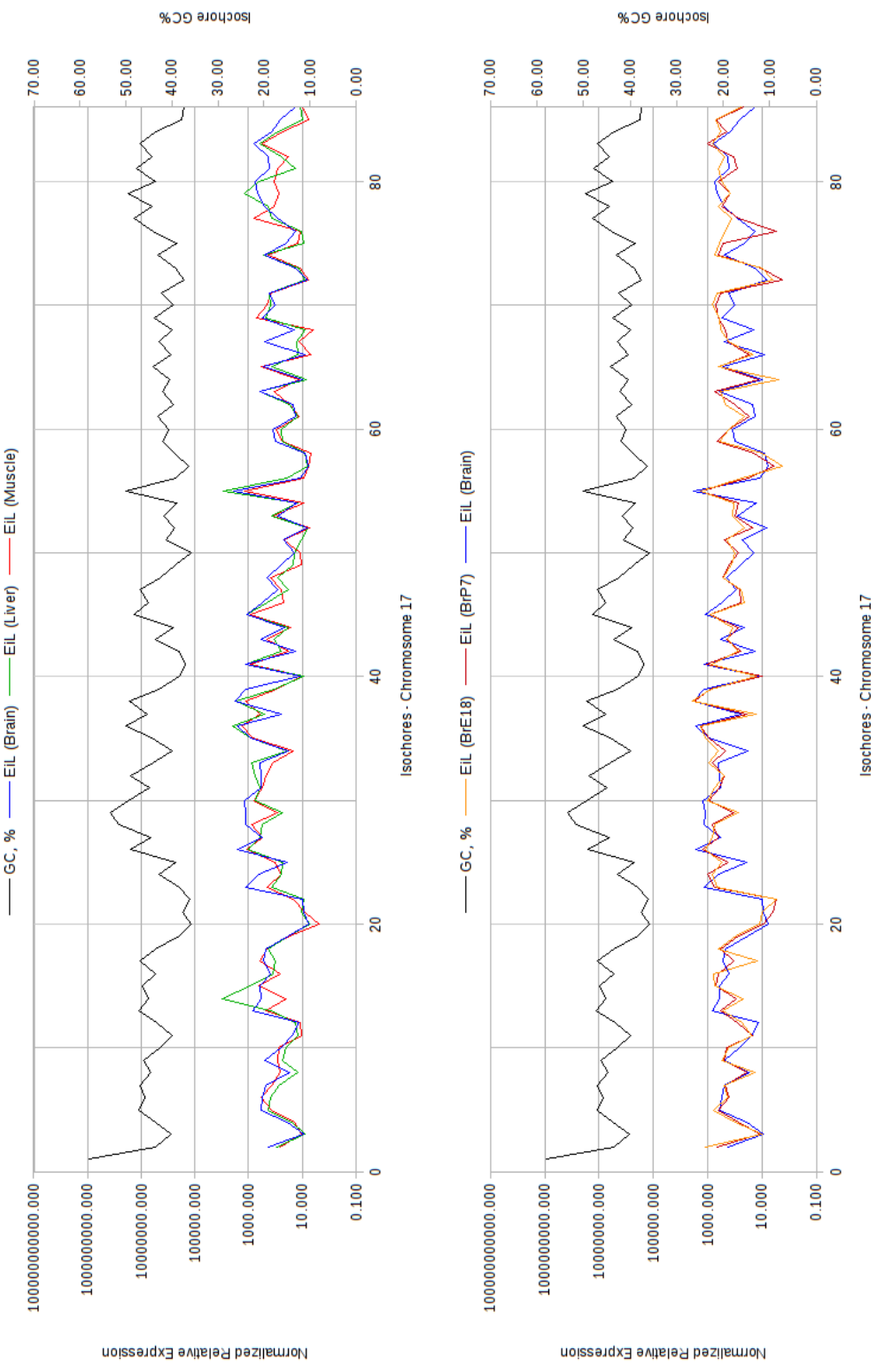


Figure A.17: *GC* content profile and expression profile along mouse chromosome 17, separately for each adult tissue and each developmental stage of the brain.

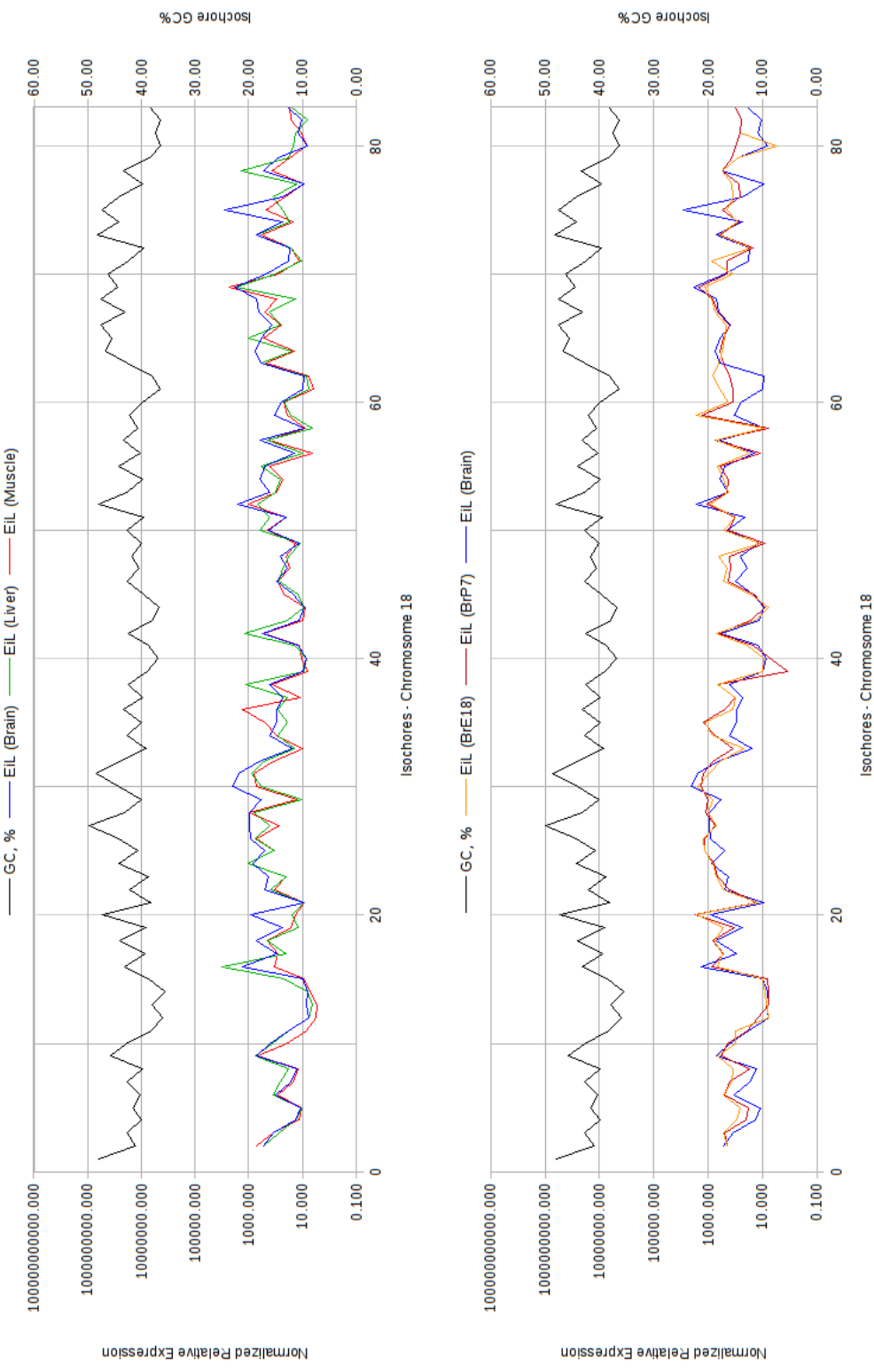


Figure A.18: *GC* content profile and expression profile along mouse chromosome 18, separately for each adult tissue and each developmental stage of the brain.

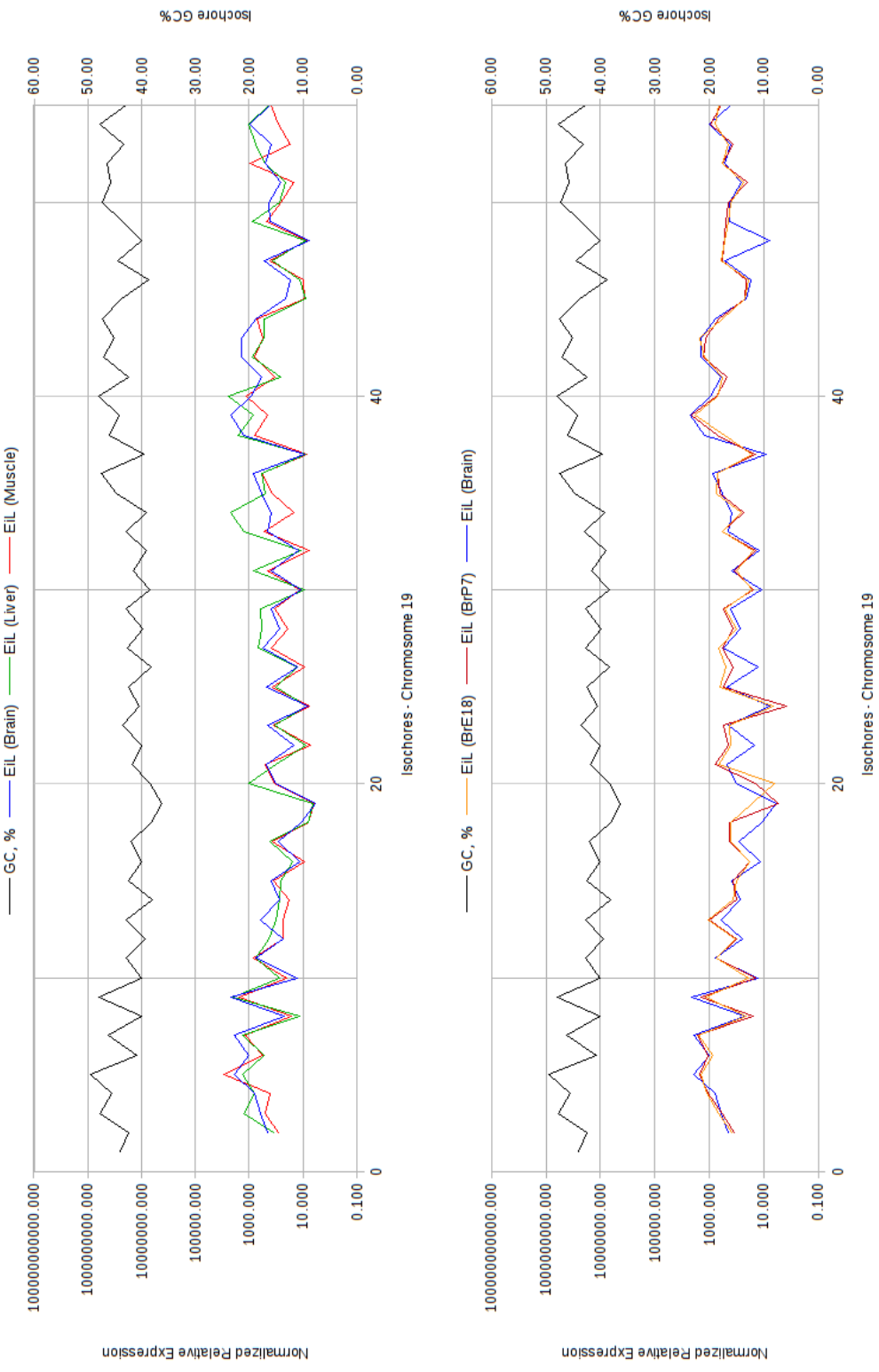


Figure A.19: *GC* content profile and expression profile along mouse chromosome 19, separately for each adult tissue and each developmental stage of the brain.

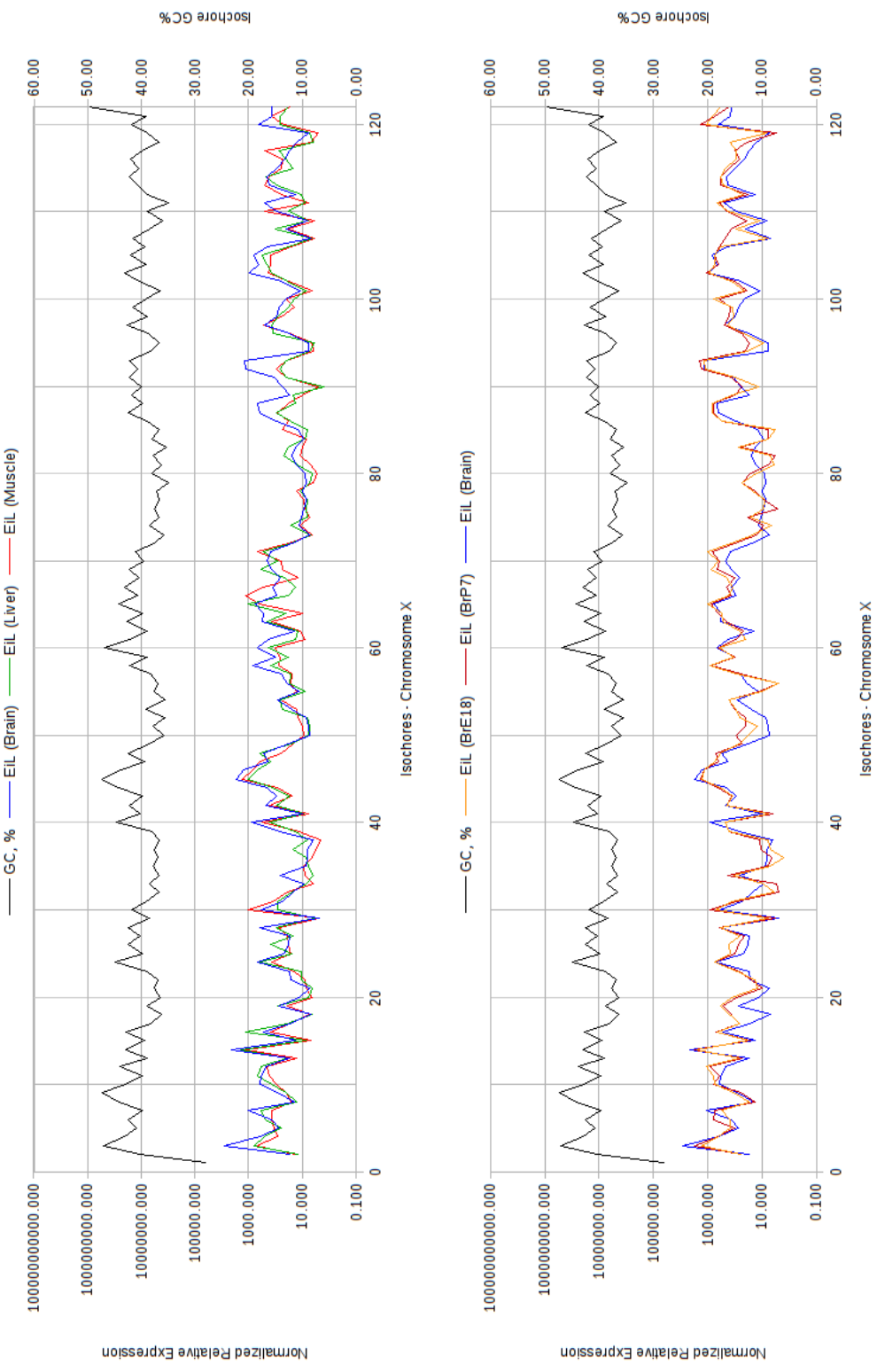


Figure A.20: *GC* content profile and expression profile along mouse chromosome X, separately for each adult tissue and each developmental stage of the brain.

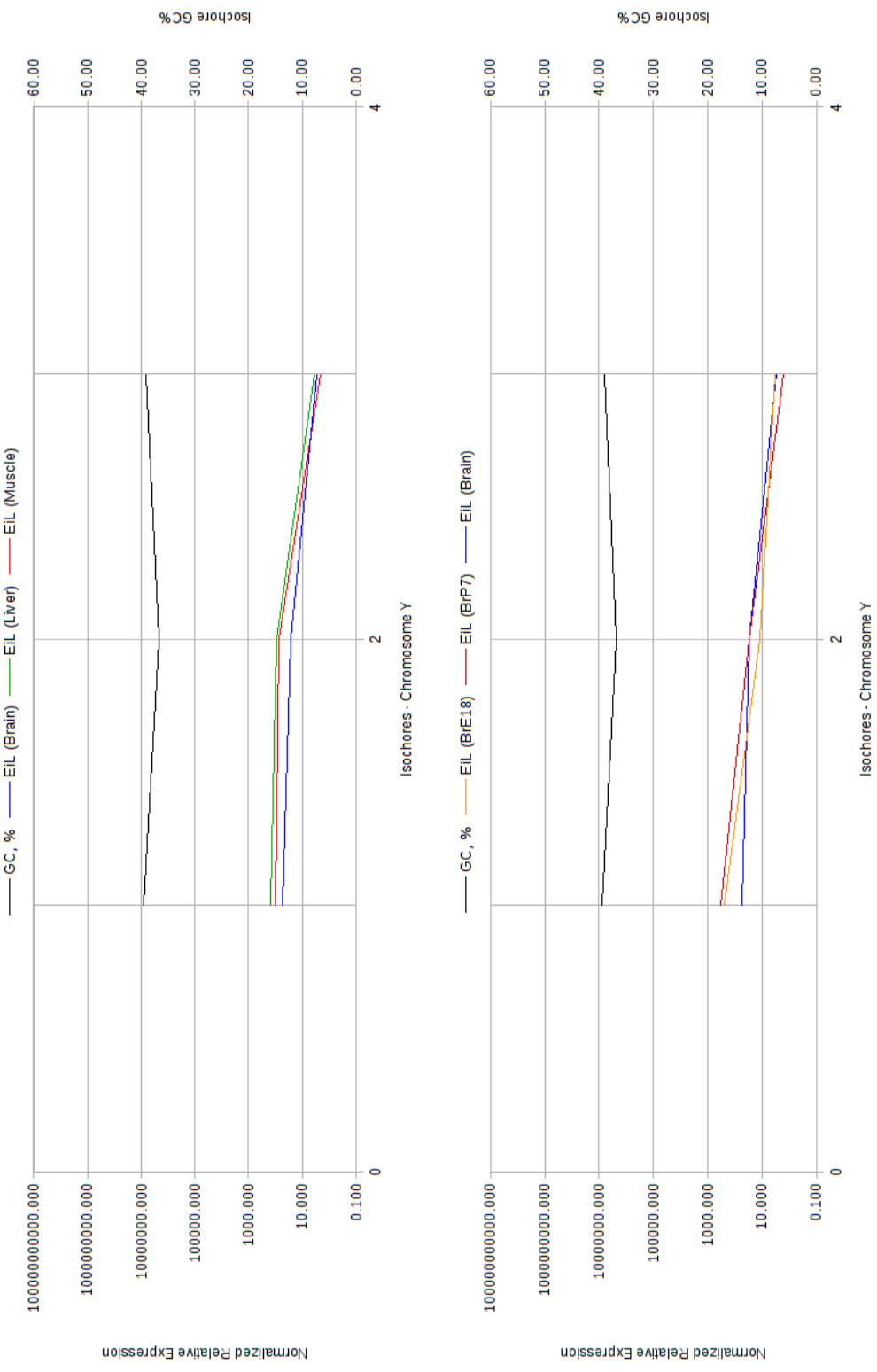


Figure A.21: *GC* content profile and expression profile along mouse chromosome Y, separately for each adult tissue and each developmental stage of the brain.

Appendix B

List of publications

[54] — REAL: An efficient REad ALigner for next generation sequencing reads

Description

REad ALigner is an efficient, accurate and consistent tool for aligning short reads obtained from next generation sequencing. It is based on a new, simple, yet efficient mapping algorithm that can match and outperform the currently fastest software, that is based on the Burrows–Wheeler Transform. The algorithm pre-process the reference and then attempts to match to it a suffix of constant given length from each read. Each suffix is allowed up to a given number of mismatches. It uses then the pigeonhole principle to split each suffix into fragments such that some of the fragments are guaranteed to have no mismatches. All possible combinations of the fragments are searched against the indexed reference to find exact matches. Then the number of mismatches in the entire suffix for each candidate alignment is counted and any matches with more mismatches than those permitted are discarded. Finally the remaining length of the read is scanned for additional mismatches, up to a maximum allowed number. The gap–less alignment with the least mismatches is reported. In case of a tie, there is an option to have either both or neither reported.

Contribution

The algorithm already existed when I joined the project, but it was not suitable for end–users. I guided its implementation into an end–user programme. I suggested searching only for a suffix from long reads, in line with the way other programs were handling long reads without affecting the efficiency of the search mechanism. I was also working on my new alignment scoring scheme for the algorithm, but I did not finish it in time for the conference.

[116] — An algorithm for mapping short reads to a dynamically changing genomic sequence

Description

The reference genome is an amalgam of the most common genomic variants. It is unlikely that any single individual will possess that exact collection of variants, thus every alignment task will face mismatches that are the result of natural variation, rather than misalignment or base call error. One way to get around this, is to take into account the known variation during the alignment stage. Given a reference and a list of single–base substitutions, the algorithm maps reads using the REAL algorithm at its core. This time the alignment reported is the one with the least mismatches across all variants.

Contribution

I proposed the idea and contributed to the general design of the algorithm, working with the computer scientists of the group to create a work-flow that would accurately address the problem.

[170] — Transcriptome map of mouse isochores

Description

Unlike usual transcriptome maps, where expression intensity is plotted against the actual position on a chromosome, this work plotted the expression against segments known as isochores, thus clustering expression activity based on how the base composition changes along a chromosome. RNAseq data from three distinct adult mouse tissues was plotted against the mouse genome isochores, revealing strong correlations between expression and base composition at isochore level. These correlations had been measured before using various techniques, but this was the first time that NGS data was used for the purpose. The alignments were performed with REAL.

Contribution

I processed the aligned data and cross-referenced it to match it to isochores and coding sequences, calculated the expression levels and helped analyse the results.

[171] — Transcriptome map of mouse isochores in embryonic and neonatal cortex

Description

This work expands on the original transcriptome map of mouse isochores, by adding two developmental stages of one of the analysed tissues.

Contribution

As before, I processed the aligned data and cross-referenced it to match it to isochores and coding sequences, calculated the expression levels and helped analyse the results.

[65] — GapMis: a Tool for Pairwise Sequence Alignment with a Single Gap

Description

The original REAL algorithm did only gap–less alignments, which is inadequate for practical use with real data, due to the presence of indels and introns. This algorithm does alignments with an upper bound on the number of allowed mismatches and up to a single gap with an upper bound on the length of that gap. The reported alignment is the one with the least Hamming distance, though the algorithm is easily modifiable to incorporate different scoring rules.

Contribution

I urged the incorporation of gapped alignments and contributed to the general design of the algorithm, as well as ensured that the algorithm would be compatible with the scoring scheme I was preparing.

[192] — Predicting the functional consequences of non–synonymous DNA sequence variants — evaluation of bioinformatics tools and development of a consensus strategy

Description

This work proposes a consensus classifier method for predicting whether a SNP has a functional consequence or not. At the time this work begun, no other consensus classifiers had been published. Two simple methods are proposed herein, both of which match and outperform the consensus classifiers that were published in the meantime.

Contribution

I came up with the idea, collected and processed all the data and implemented the methods and their interfaces.

List of Figures

List of Tables

Bibliography

[1] J. ten Bosch and W. Grody, "Keeping Up With the Next Generation: Massively Parallel Sequencing in Clinical Diagnostics," *Journal of Molecular Diagnostics*, vol. 10, p. 484, November 2008.

[2] H. Buermans and J. den Dunnen, "Next generation sequencing: Advaces and applications," *Biochimica et Biophysica Acta*, vol. 1842, p. 1932, 2014.

[3] The 1000 Genomes Project Consortium, "A map of human genome variation from population–scale sequencing.," *Nature*, vol. 467, p. 1061, 2010.

[4] D. Bentley, S. Balasubramanian, H. Swerdlow, *et al.*, "Accurate whole genome sequencing using reversible terminator chemistry," *Nature*, vol. 456, no. 7218, p. 53, 2008.

[5] M. Margulies, M. Egholm, W. Altman, *et al.*, "Genome sequencing in open microfabricated high density picoliter reactors," *Nature*, vol. 437, p. 376, 2005.

[6] B. Merriman, Ion Torrent R&D Team, and J. Rothberg, "Progress in ion torrent semiconductor chip based sequencing," *Electrophoresis*, vol. 33, no. 23, p. 3397, 2012.

[7] J. Shendure, G. Porreca, N. Reppas, *et al.*, "Accurate multiplex polony sequencing of an evolved bacterial genome," *Science*, vol. 309, no. 5741, p. 1728, 2005.

[8] D. Aird, M. Ross, W. Chen, *et al.*, "Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries," *Genome Biology*, vol. 12, p. R18, 2011.

[9] J. Thompson and K. Steinmann, *Single molecule sequencing with a HeliScope genetic analysis system*, ch. 7.10. Wiley, 2010.

[10] A. Dalca and M. Brudno, "Genome variation discovery with high-throughput sequencing data," *Briefings in Bioinformatics*, vol. 11, no. 1, p. 3, 2010.

[11] P. Lundquist, C. Zhong, P. Zhao, *et al.*, "Parallel confocal detection of single molecules in real time," *Opt Lett*, vol. 33, no. 9, p. 1026, 2008.

[12] J. Eid, A. Fehr, J. Gray, *et al.*, "Real-time DNA sequencing from single polymerase molecules," *Science*, vol. 323, no. 5910, p. 133, 2009.

[13] N. Ashkenasy, J. S'anchez-Quesada, and M. Ghadiri, "Recognising a single base in an individual DNA strand: a step toward nanopore DNA sequencing," *Angew Chem Int Ed Engl*, vol. 44, no. 9, p. 1401, 2005.

[14] Y. Astier, B. O., and H. Bayley, "Toward single molecule DNA sequencing: direct identification of ribonucleoside and deoxyribonucleoside 5'-monophosphates by using an engineered protein nanopore equipped with a molecular adapter," *J Am Chem Soc*, vol. 128, no. 5, p. 1705, 2006.

[15] J. Clarke, H. Wu, L. Jayasinghe, *et al.*, "Continuous base identification for single-molecule nanopore DNA sequencing," *Nat Nanotechnol*, vol. 4, no. 4, p. 265, 2009.

[16] M. Tsutsui, M. Taniguchi, K. Yokota, and T. Kawai, "Identifying single nucleotides by tunnelling current," *Nat Nanotechnol*, vol. 5, no. 4, p. 286, 2010.

[17] Y. Erlich, P. Mitra, M. delaBastide, *et al.*, "Alta-Cyclic: a self-optimizing base caller for next-generation sequencing," *Nat Methods*, vol. 5, no. 8, p. 679, 2008.

[18] J. Rougemont, A. Amzallag, C. Iseli, *et al.*, "Probabilistic base calling of Solexa sequencing data," *BMC Bioinformatics*, vol. 9, p. 431, 2008.

[19] B. Ewing, L. Hillier, M. Wendl, and P. Green, "Base-Calling of Automated Sequencer Traces Using Phred. I. Accuracy assessment," *Genome Research*, vol. 8, no. 3, p. 175, 1998.

[20] B. Ewing and P. Green, "Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities," *Genome Research*, vol. 8, p. 186, March 1998.

[21] N. Malhis, Y. Butterfield, M. Ester, and S. Jones, "Slider—maximum use of probability information for alignment of short sequence reads and SNP detection," *Bioinformatics*, vol. 25, p. 6, January 2009.

[22] W. Kao, K. Stevens, and Y. Song, "BayesCall: A model-based base-calling algorithm for high-throughput short-read sequencing," *Genome Research*, vol. 19, no. 10, p. 1884, 2009.

[23] W. Kao and Y. Song, "naiveBayesCall: an efficient model-based base-calling algorithm for high-throughput sequencing," *J Comput Biol*, vol. 13, no. 3, p. 365, 2011.

[24] S. Das and H. Vikalo, "OnlineCall: fast online estimation and base calling fo illumina's next-generation sequencing," *Bioinformatics*, vol. 28, no. 13, p. 1677, 2012.

[25] S. Das and H. Vikalo, "Base calling for high-throughput short-read sequencing: dynamic programmin solutions.," *BMC Bioinformatics*, vol. 14, p. 129, 2013.

[26] H. Bravo and R. Irizarry, "Model-based quality assessment and base-calling for second-generation sequencing data," *Biometrics*, vol. 66, no. 3, p. 665, 2010.

[27] T. Massingham and G. N., "All your Base: a fast and accurate probabilistic approach to base calling," *Genome Biology*, vol. 13, no. 2, p. R13, 2012.

[28] C. Ye, C. Hsiao, and H. Bravo, "BlindCall ultra-fast base-calling of high-throughput sequencing data by blind deconvolution.," *Bioinformatics*, vol. 30, no. 9, p. 1214, 2014.

[29] M. Kircher, U. Stenzel, and J. Kelso, "Improved base calling for the Illumina Genome Analyzer using machine learning strategies," *Genome Biology*, vol. 10, no. 8, p. R83, 2009.

[30] G. Renaud, M. Kircher, U. Stenzel, and J. Kelso, "freeIbis: an efficient basecaller with calibrated quality scores for Illumina sequencers," *Bioinformatics*, vol. 29, no. 9, p. 1208, 2013.

[31] C. Ledergerber and C. Dessimoz, "Base-Calling for next-generation sequencing platforms," *Briefings in Bioinformatics*, vol. 12, no. 5, p. 489, 2011.

[32] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, p. 443, 1970.

[33] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, p. 195, March 1981.

[34] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proceedings of the National Academy of Sciences*, vol. 85, p. 2444, 1988.

[35] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic Local Aignment Search Tool," *Journal of Molecular Biology*, vol. 215, p. 403, October 1990.

[36] W. Kent, "BLAT—The BLAST-Like Alignment Tool," *Genome Research*, vol. 12, p. 656, April 2002.

[37] K. Kalafus, A. Jackson, and A. Milosavljevic, "Pash: efficient genome-scale sequence anchoring by Positional Hashing," *Genome Research*, vol. 14, p. 672, April 2004.

[38] A. Cox, "ELAND." 2005.

[39] X. Zhang, Z. Cao, Z. Lin, Q. Wang, and Y. Li, "EMMA: An Efficient Massive Mapping Algorithm Using Improved Approximate Mapping Filtering," *Acta Biochimica et Biophysica Sinica*, vol. 38, p. 857, December 2006.

[40] C. Coarfa and A. Milosavljevic, "Pash 2.0: scaleable sequence anchoring for next-generation sequencing technologies," *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, p. 102, 2008.

[41] D. Campagna, A. Albiero, A. Bilardi, E. Caniato, C. Forcato, S. Manavski, N. Vitulo, and G. Valle, "PASS: a program to align short sequences," *Bioinformatics*, vol. 25, p. 967, April 2009.

[42] H. Li, J. Ruan, and R. Durbin, "Mapping short DNA sequencing reads and calling variants using mapping quality scores," *Genome Research*, vol. 18, p. 1851, November 2008.

[43] H. Jiang and W. Wong, "SeqMap: mapping massive amount of oligonucleotides to the genome," *Bioinformatics*, vol. 24, p. 2395, October 2008.

[44] R. Li, Y. Li, K. Kristiansen, and J. Wang, "SOAP: short oligonucleotide alignment program," *Bioinformatics*, vol. 24, p. 713, January 2008.

[45] H. Eaves and Y. Gao, "MOM: Maximum Oligonucleotide Mapping," *Bioinformatics*, vol. 25, p. 969, April 2009.

[46] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows-Wheeler transform," *Bioinformatics*, vol. 25, p. 1754, July 2009.

[47] N. Homer, B. Merriman, and S. Nelson, "BFAST: An Alignment Tool for Large Scale Genome Resequencing," *PLoS ONE*, vol. 4, p. e7767, November 2009.

[48] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome Biology*, vol. 10, no. 3, p. R25, 2009.

[49] R. Li, C. Yu, Y. Li, T. Lam, S. Yiu, K. Kristiansen, and J. Wang, "SOAP2: an improved ultrafast tool for short read alignment," *Bioinformatics*, vol. 25, p. 1966, August 2009.

[50] K. Schneeberger, J. Hagmann, S. Ossowski, N. Warthmann, S. Gesing, O. Kohlbacher, and D. Weigel, "Simultaneous alignment of short reads against multiple genomes," *Genome Biology*, vol. 10, p. R98, September 2009.

[51] W. Wang, P. Zhang, and X. Liu, "Short read DNA fragment anchoring algorithm," *BMC Bioinformatics*, vol. 10, no. Suppl 1, p. S17, 2009.

[52] J. Na, K. Roh, A. Apostolico, and K. Park, "Alignment of biological sequences with quality scores," *International Journal for Bioinformatics Research and Applicaitons*, vol. 5, no. 1, p. 97, 2009.

[53] P. Antoniou, C. Iliopoulos, L. Mouchard, and S. Pissis, *A fast and efficient algorithm for mapping short sequences to a reference genome*, vol. Advances in Experimental Medicine and Biology. Springer, 2010.

[54] K. Frousios, C. Iliopoulos, L. Mouchard, S. Pissis, and G. Tischler, "REAL: An efficient REad ALigner for next generation sequencing reads,"

in *Proceedings of the first ACM International Conference on Bioinformatics and Computational Biology*, BCB '10, (New York, NY, USA), p. 154, ACM, 2010.

[55] D. Jaffe, J. Butler, S. Gnerre, *et al.*, "Whole-genome sequence assembly for mammalian genomes: Arachne 2," *Genome Research*, vol. 13, no. 1, p. 91, 2003.

[56] R. Warren, G. Sutton, S. Jones, and R. Holt, "Assembling millions of short DNA sequences using SSAKE," *Bioinformatics*, vol. 23, p. 500, February 2007.

[57] W. Jeck, J. Reinhardt, D. Baltrus, M. Hickenbotham, V. Magrini, E. Mardis, J. Dangl, and C. Jones, "Extending assembly of short DNA sequences to handle error," *Bioinformatics*, vol. 23, p. 2942, November 2007.

[58] J. Butler, I. MacCallum, M. Kleber, *et al.*, "ALLPATHS: De novo assembly of whole-genome shotgun microreads," *Genome Research*, vol. 18, no. 5, p. 810, 2008.

[59] M. Chaisson and P. Pevzner, "Short read fragment assembly of bacterial genomes," *Genome Research*, vol. 18, no. 2, p. 324, 2008.

[60] D. Zerbino and E. Birney, "Velvet: Algorithms for de novo short read assembly using de Bruijn graphs," *Genome Research*, vol. 18, p. 821, May 2008.

[61] J. Simpson, K. Wong, S. Jackman, J. Schein, S. Jones, and I. Birol, "ABySS: a parallel assembler for short read sequence data," *Genome Research*, vol. 19, p. 1117, June 2009.

[62] R. Li, H. Zhu, J. Ruan, *et al.*, "De novo assembly of human genomes with massively parallel short read sequencing," *Genome Research*, vol. 20, p. 265, February 2010.

[63] P. Pevzner, H. Tang, and M. Waterman, "An Eulerian path approach to DNA fragment assembly," *PNAS*, vol. 98, p. 9748, August 2001.

[64] C. Alkan, S. Sajjadian, and E. Eichler, "Limitations of next-generation genome sequence assembly," *Nat Methods*, vol. 8, no. 1, p. 61, 2011.

[65] T. Flouri, K. Frousios, C. Iliopoulos, *et al.*, "GapMis: a Tool for Pairwise Sequence Alignment with a Single Gap," *Recent Patents on DNA & Gene Sequences*, vol. 7, no. 2, p. 84, 2013.

[66] A. McKenna, M. Hanna, E. Banks, *et al.*, "The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data," *Genome Research*, vol. 20, p. 1297, 2010.

[67] H. Li, B. Handsaker, A. Wysoker, *et al.*, "The Sequence Alignment/Map format and SAMtools," *Bioinformatics*, vol. 25, p. 2078, August 2009.

[68] D. Challis, J. Yu, U. E vani, *et al.*, "An integrative variant analysis suite for whole exome next-generation sequencing data," *BMC Bioinformatics*, vol. 13, p. 8, 2012.

[69] X. Liu, S. Han, Z. Wang, *et al.*, "Variant callers for next-generation sequencing data: a comparison study," *PLoS One*, vol. 8, no. 9, p. e75619, 2013.

[70] R. Li, Y. Li, X. Fang, *et al.*, "SNP detection for massively parallel whole–genome resequencing," *Genome Research*, vol. 19, no. 6, p. 1124, 2009.

[71] K. Holt, Y. Teo, H. Li, S. Nair, G. Dougan, J. Wain, and J. Parkhill, "Detecting SNPs and estimating allele frequencies in clonal bacterial populations by sequencing pooled DNA," *Bioinformatics*, vol. 25, p. 2074, August 2009.

[72] B. Langmead, M. Schatz, J. Lin, *et al.*, "Searching for SNPs with cloud computing.," *Genome Biology*, vol. 10, p. R134, 2009.

[73] H. Chang, L. Chuang, Y. Cheng, C. Ho, C. Wen, and C. Yang, "Seq-SNPing: Multiple-Alignment Tool for SNP Discovery, SNP ID Identification, and RFLP Genotyping," *OMICS: A Journal of Integrative Biology*, vol. 13, p. 253, June 2009.

[74] E. Martin, D. Kinnamon, M. Schmidt, *et al.*, "SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies," *Bioinformatics*, vol. 26, no. 22, p. 2803, 2010.

[75] X. Yu and S. Sun, "Comparing a few SNP algorithms using low-coverage sequencing data," *BMC Bioinformatics*, vol. 14, p. 274, 2013.

[76] P. Ng and S. Henikoff, "SIFT: predicting amino acid changes that affect protein function," *Nucleic Acids Research*, vol. 31, no. 13, p. 3812, 2003.

[77] I. Adzhubei, S. Schmidt, L. Peshkin, *et al.*, "A method and server for predicting damaging missense mutations," *Nature Methods*, vol. 7, no. 4, p. 248, 2010.

[78] B. Li, V. Krishnan, M. Mort, *et al.*, "Automated inference of molecular mechanisms of disease from amino acid substitutions," *Bioinformatics*, vol. 25, no. 21, p. 2744, 2009.

[79] R. Calabrese, E. Capriotti, P. Fariselli, *et al.*, "Functional annotations improve the predictive score of human disease-related mutations in proteins," *Human Mutation*, vol. 30, p. 1237, 2009.

[80] P. Thomas, A. Kejariwal, N. Guo, *et al.*, "Applications for protein sequence-function evolution data: mRNA/protein expression analysis and coding SNP scoring tools," *Nucleic Acids Research*, vol. 34, no. Web Server Issue, p. W645, 2006.

[81] E. Capriotti, R. Calabrese, and R. Casadio, "Predicting the insurgence of human genetic diseases associated to single point protein mutations with

support vector machines and evolutionary information," *Bioinformatics*, vol. 22, no. 22, p. 2729, 2006.

[82] B. Reva, Y. Antipin, and C. Sander, "Predicting the functional impact of protein mutations: application to cancer genomics," *Nucleic Acids Research*, vol. 39, no. 17, p. e118, 2011.

[83] L. Bao, M. Zhou, and Y. Cui, "nsSNPAnalyzer: identifying disease-associated nonsynonymous single nucleotide polymorphisms," *Nucleic Acids Research*, vol. 33, no. Web Server Issue, p. W480, 2005.

[84] M. Lopes, C. Joyce, G. Ritchie, *et al.*, "A combined functional annotation score for non-synonymous variants.," *Hum Hered*, vol. 73, no. 1, p. 47, 2012.

[85] M. Barenboim, M. Masso, I. Vaisman, and D. Jamison, "Statistical geometry based prediction of nonsynonymous SNP functional effects using random forest and neuro-fuzzy classifiers," *Proteins: Structure, Function and Bioinformatics*, vol. 71, no. 4, p. 1930, 2008.

[86] Y. Bromberg and B. Rost, "SNAP: predict effect of non-synonymous polymorphisms an function," *Nucleic Acids Research*, vol. 35, no. 11, p. 3823, 2007.

[87] E. Capriotti and R. Altman, "Improving the prediction of disease-related variants using protein three-dimesional structure," *BMC Bioinformatics*, vol. 12, no. Suppl 4, p. S3, 2011.

[88] C. Chelala, A. Khan, and N. Lemoine, "SNPnexus: a web database for functional annotation of newly discovered and public domain single nucleotide polymorphisms," *Bioinformatics*, vol. 25, no. 5, p. 655, 2008.

[89] L. Conde, J. Vaquerizas, H. Dopazo, *et al.*, "PupaSuite: finding functional single nucleotide polymorphisms for large–scale genotyping pur-

poses," *Nucleic Acids Research*, vol. 34, no. Web Server Issue, p. W621, 2006.

[90] J. Dantzer, C. Moad, R. Heiland, and S. Mooney, "MutDB services: interactive structural analysis of mutation data," *Nucleic Acids Research*, vol. 33, no. Web Server Issue, p. W311, 2005.

[91] C. Ferrer-Costa, J. Gelpi, L. Zamakola, *et al.*, "PMUT: a web-based tool for the annotation of pathological mutations of proteins," *Bioinformatics*, vol. 21, no. 14, p. 3176, 2005.

[92] A. Gonzalez-Perez and N. Lopez-Bigas, "Improving the assessment of the outcome of nonsynonymous SNVs with a consensus deleteriousness score, Condel," *The American Journal of Human Genetics*, vol. 88, p. 440, 4 2011.

[93] B. Hemminger, B. Saelim, and F. Sullivan, "TAMAL: an integrated approach to choosing SNPs for genetic studies of human complex traits," *Bioinformatics*, vol. 22, no. 5, p. 626, 2006.

[94] H. Kang, K. Choi, B. Kim, *et al.*, "FESD: a Functional Element SNPs Database in human," *Nucleic Acids Research*, vol. 33, no. Database Issue, p. D518, 2005.

[95] R. Karchin, M. Diekhans, L. Kelly, *et al.*, "LS-SNP: large-scale annotation of coding non-synonymous SNPs based on multiple information sources," *Bioinformatics*, vol. 21, no. 12, p. 2814, 2005.

[96] P. Lee and H. Shatkay, "F–SNP: computationally predicted functional SNPs for disease association studies," *Nucleic Acids Research*, vol. 36, no. Database Issue, p. D820, 2008.

[97] W. McLaren, B. Pritchard, D. Rios, *et al.*, "Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor," *Bioinformatics*, vol. 26, no. 16, p. 2069, 2010.

[98] J. Reumers, S. Maurer-Stroh, J. Schymkowitz, and F. Rousseau, "SNPeffect v2.0: a new step in investigating the molecular phenotypic effects of human non-synonymousSNPs," *Bioinformatics*, vol. 22, no. 17, p. 2183, 2006.

[99] J. Thusberg and M. Vihinen, "Pathogenic or not? And if so, then how? Studying the effects of missense mutations using Bioinformatics Mehtods," *Human Mutation*, vol. 30, p. 703, 2009.

[100] K. Wang, M. Li, and H. Hakonarson, "ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data," *Nucleic Acids Research*, vol. 38, no. 16, p. e164, 2010.

[101] H. Yuan, J. Chiou, W. Tseng, *et al.*, "FASTSNP: an always up–to–date and extendable service for SNP function analysis and prioritization," *Nucleic Acids Research*, vol. 34, no. Web Server Issue, p. W635, 2006.

[102] P. Yue, E. Melamud, and J. Moult, "SNPs3D: Candidate gene and SNP selection for association studies," *BMC Bioinformatics*, vol. 7, p. 166, 2006.

[103] F. Ozsolak and P. Milos, "RNA sequencing: advances, challenges and opportunities," *Nature Reviews Genetics*, vol. 12, no. 2, p. 87, 2011.

[104] S. Sarda and S. Hannenhalli, "Next-generation sequencing and epigenomics research: a hammer in search for nails," *Genomics Inform*, vol. 12, no. 1, p. 2, 2014.

[105] J. Miller, S. Koren, and G. Sutton, "Assembly algorithms for next-generation sequencing data," *Genomics*, vol. 95, p. 315, June 2010.

[106] K. Malde, "The effect of sequence quality on sequence alignment," *Bioinformatics*, vol. 24, no. 7, p. 897, 2008.

[107] M. Frith, R. Wan, and P. Horton, "Incorporating sequence quality data into alignment improves DNA read mapping," *Nucleic Acids Research*, vol. 38, no. 7, p. e100, 2010.

[108] S. Karlin and S. Altschul, "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes," *PNAS*, vol. 87, no. 6, p. 2264, 1990.

[109] M. Dayhoff, R. Schwartz, and B. Orcutt, "Chapter 22: A model of evolutionary change in proteins," in *Atlas of Protein Sequence and Structure* (M. O. Dayhoff, ed.), pp. 345+, 1978.

[110] S. Henikoff and J. Henikoff, "Amino acid substitution matrices from protein blocks," *PNAS*, vol. 89, no. 22, p. 10915, 1992.

[111] J. Schneider, M. Pungliya, J. Choi, R. Jiang, X. Sun, B. Salisbury, and J. Stephens, "DNA variability of human genes," *Mechanisms of Ageing and Development*, vol. 124, no. 1, p. 17, 2003. Functional Genomics of Ageing I.

[112] S. Altschul, J. Wootton, E. Zaslavsky, and Y. Yu, "The Construction and Use of Log-Odds Substitution Scores for Multiple Sequence Alignment," *PLoS Computational Biology*, vol. 6, p. e1000852, July 2010.

[113] D. Benson, I. Karsch-Mizrachi, D. Lipman, *et al.*, "GenBank," *Nucleic Acids Research*, vol. 39, no. Database issue, p. D32, 2011.

[114] F. Ozsolak, "Third generation sequencing techniques and applications to drug discovery," *Expert Opinion in Drug Discovery*, vol. 7, no. 3, p. 231, 2012.

[115] *Mapping short reads to a genomic sequence with circular structure*, 2010.

[116] T. Flouri, J. Holub, C. Iliopoulos, *et al.*, "An algorithm for mapping short reads to a dynamically changing genomic sequence," in *Proceedings of the*

International Conference on Bioinformatics & Biomedicine (BIBM2010), p. 133, 2010.

[117] J. Thiery, G. Macaya, and G. Bernardi, "An analysis of eukaryotic genomes by density gradient centrifugation," *J Mol Biol*, vol. 108, no. 1, p. 219, 1976.

[118] G. Bernardi, B. Olofsson, J. Filipski, *et al.*, "The mosaic genome of warm-blooded vertebrates," *Science*, vol. 228, p. 953, 1985.

[119] M. Costantini, R. Cammarano, and G. Bernardi, "The evolution of isochore patterns in vertebrate genomes," *BMC Genomics*, vol. 10, no. 1, p. 146, 2009.

[120] T. Fukagawa, K. Sugaya, K. Matsumoto, *et al.*, "A boundary of long-range G + C% mosaic domains in the human MHC locus: pseudoautosomal boundary-like sequence exists near the boundary.," *Genomics*, vol. 25, no. 1, p. 184, 1995.

[121] A. De Sario, E. Geigl, G. Palmieri, *et al.*, "A compositional map of human chromosome band Xq28," *Proceedings of the National Academy of Sciences USA*, vol. 93, no. 3, p. 1298, 1996.

[122] A. De Sario, G. Roizes, N. Allegre, and G. Bernardi, "A compositional map of the cen-q21 region of human chromosome 21," *Gene*, vol. 194, no. 1, p. 107, 1997.

[123] M. Lercher, A. Urrutia, A. Pavlicek, and L. Hurst, "A unification of mosaic structures in the human genome," *Human Molecular Genetics*, vol. 12, no. 19, p. 2411, 2003.

[124] D. Mouchiroud, G. D'Onofrio, B. Aïssani, *et al.*, "The distribution of genes in the human genome," *Gene*, vol. 100, p. 181, 1991.

[125] S. Zoubak, O. Clay, and G. Bernardi, "The gene distribution of the human genome," *Gene*, vol. 174, no. 1, p. 95, 1996.

[126] R. Versteeg, B. van Schaik, M. van Batenburg, *et al.*, "The human transcriptome map reveals extremes in gene density, intron length, GC content, and repeat pattern for domains of highly and weakly expressed genes," *Genome Research*, vol. 13, no. 9, p. 1998, 2003.

[127] H. Caron, B. van Schaik, M. van der Mee, *et al.*, "The Human Transcriptome Map: Clustering of Highly Expressed Genes in Chromosomal Domains," *Science*, vol. 291, no. 5507, p. 1289, 2001.

[128] C. Federico, S. Saccone, and G. Bernardi, "The gene-richest bands of human chromosomes replicate at the onset of the S-phase.," *Cytogenetics and Cell Genetics*, vol. 80, p. 83, 1998.

[129] S. Fullerton, A. Bernardo Carvalho, and A. Clark, "Local rates of recombination are positively correlated with GC content in the human genome," *Molecular Biology and Evolution*, vol. 18, no. 6, p. 1139, 2001.

[130] S. Saccone, A. Desario, J. Wiegant, *et al.*, "Correlations between isochores and chromosomal bands in hte human genome.," *Proceedings of the National Academy of Sciences USA*, vol. 90, p. 11929, 1993.

[131] C. Federico, L. Andreozzi, S. Saccone, and G. Bernardi, "Gene density in the Giemsa bands of human chromosomes," *Chromosome Research*, vol. 8, p. 737, 2000.

[132] I. Hiratani, A. Leskovar, and D. Gilbert, "Differentiation-induced replication-timing changes are restricted to AT-rich/long interspersed nuclear element (LINE)-rich isochores," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 48, p. 16861, 2004.

[133] L. Ren, G. Gao, D. Zhao, *et al.*, "Developmental stage related patterns of codon usage and genomic GC content: Searching for evolutionary fingerprints with models of stem cell differentiation," *Genome Biology*, vol. 8, no. 3, p. R35, 2007. cited By (since 1996) 5.

[134] O. Clay and G. Bernardi, "GC3 of Genes Can Be Used as a Proxy for Isochore Base Composition: A Reply to Elhaik et al.," *Molecular Biology and Evolution*, vol. 28, no. 1, p. 21, 2011.

[135] A. Vinogradov, "Isochores and tissue specificity," *Nucleic Acids Research*, vol. 31, no. 17, p. 5212, 2003.

[136] G. Pesole, G. Bernardi, and C. Saccone, "Isochore specificity of AUG initiator context of human genes," *FEBS Letters*, vol. 464, no. 1–2, p. 60, 1999.

[137] P. Carninci, A. Sandelin, B. Lenhard, *et al.*, "Genome-wide analysis of mammalian promoter architecture and evolution," *Nature Genetics*, vol. 38, no. 6, p. 626, 2006.

[138] V. Bajic, S. Tan, A. Christoffels, *et al.*, "Mice and Men: Their Promoter Properties," *PLoS Genet*, vol. 2, p. e54, 04 2006.

[139] G. D'Onofrio, T. Ghosh, and S. Saccone, "Different functional classes of genes are characterized by different compositional properties," *FEBS Letters*, vol. 581, no. 30, p. 5819, 2007.

[140] S. Arhondakis, F. Auletta, and G. Bernardi, "Isochores and the Regulation of Gene Expression in the Human Genome," *Genome Biology and Evolution*, vol. 3, p. 1080, 2011.

[141] G. Bernardi, "The neoselectionist Theory of Genome Evolution," *PNAS*, vol. 104, no. 20, p. 8385, 2007.

[142] L. Duret and N. Galtier, "Biased Gene Conversion and the Evolution of Mammalian Genomic Landscapes," *Annual Review of Genomics and Human Genetics*, vol. 10, no. 1, p. 285, 2009.

[143] G. Bernardi, "The genome: an isochore ensemble and its evolution," *Annals of the New York Academy of Sciences*, vol. 1267, p. 31, 2012.

[144] K. Wolfe, S. Myers, D. Richter, *et al.*, "Mutation rates differ among regions of the mammalian genome," *Nature*, vol. 337, p. 283, 1989.

[145] N. Galtier, G. Piganeau, D. Mouchiroud, and L. Duret, "GC–Content Evolution in Mammalian Genomes: The Biased Gene Conversion Hypothesis," *Genetics*, vol. 159, no. 2, p. 907, 2001.

[146] D. Häring and J. Kypr, "No isochores in the human genome chromosomes 21 and 22?," *Biochemical and Biophysical Research Communications*, vol. 280, p. 567, 2001.

[147] N. Cohen, T. Dagan, L. Stone, and D. Graur, "GC composition of the human genome: in search of isochores," *Mol. Biol. Evol.*, vol. 22, no. 5, p. 1260, 2005.

[148] E. Elhaik, D. Graur, K. Josić, and G. Landan, "Identifying compositionally homogeneous and nonhomogeneous domains within the human genome using a novel segmentation algorithm," *Nucleic Acids Research*, vol. 38, no. 15, p. e158, 2010.

[149] M. Costantini, F. Alvarez-Valin, S. Costantini, *et al.*, "Compositional patterns in the genomes of unicellular eukaryotes.," *BMC Genomics*, vol. 14, p. 755, 2013.

[150] R. Ream, G. Johns, and G. Somero, "Base compositions of genes encoding alpha-actin and lactate dehydrogenase-A from differently adapted vertebrates show no temperature-adaptive variation in G + C content," *Mol. Biol. Evol.*, vol. 20, p. 105, 2003.

[151] M. Belle, L. Duret, N. Galtier, and A. Eyre-Walker, "The decline of isochores in mammals: An assessment of the GC content variation along the mammalian phylogeny," *Journal of Molecular Evolution*, vol. 580, p. 653, 2004.

[152] A. Eyre-Walker and L. Hurst, "The evolution of isochores," *Nature Reviews Genetics*, vol. 2, no. 7, p. 549, 2001.

[153] L. Duret, "Evolution of synonymous codon usage in metazoans," *Current Opinion in Genetics & Development*, vol. 12, no. 6, p. 640, 2002.

[154] O. Konu and M. Li, "Correlations between mRNA expression levels and GC contents of coding and untranslated regions of genes in rodents," *Journal of Molecular Evolution*, vol. 54, no. 1, p. 35, 2002.

[155] S. Arhondakis, F. Auletta, G. Torelli, and G. D'Onofrio, "Base composition and expression level of human genes," *Gene*, vol. 325, p. 165, 2004.

[156] J. Comeron, "Selective and Mutational Patterns Associated With Gene Expression in Humans: Influences on Synonymous Composition and Intron Presence," *Genetics*, vol. 167, no. 3, p. 1293, 2004.

[157] M. Semon, D. Mouchiroud, and L. Duret, "Relationship between gene expression and GC-content in mammals: statistical significance and biological relevance," *Human Molecular Genetics*, vol. 14, p. 421, February 2005.

[158] A. Vinogradov, "Dualism of gene GC content and CpG pattern in regard to expression in the human genome: Magnitude versus breadth," *Trends in Genetics*, vol. 21, no. 12, p. 639, 2005.

[159] S. Arhondakis, O. Clay, and G. Bernardi, "Compositional properties of human cDNA libraries: Practical implications," *FEBS Letters*, vol. 580, no. 24, p. 5772, 2006.

[160] S. Arhondakis, O. Clay, and G. Bernardi, "GC level and expression of human coding sequences," *Biochemical and Biophysical Research Communications*, vol. 367, no. 3, p. 542, 2008.

[161] T. Mijalski, A. Harder, T. Halder, *et al.*, "Identification of coexpressed gene clusters in a comparative analysis of transcriptome and proteome in mouse tissues," *PNAS*, vol. 102, no. 24, p. 8621, 2005.

[162] G. Singer, A. Lloyd, L. Huminiecki, and K. Wolfe, "Clusters of Co-expressed Genes in Mammalian Genomes Are Conserved by Natural Selection," *Molecular Biology and Evolution*, vol. 22, no. 3, p. 767, 2005.

[163] A. Mortazavi, B. Williams, K. McCue, *et al.*, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nature Methods*, vol. 5, p. 621, 7 2008.

[164] X. Han, X. Wu, W. Chung, *et al.*, "Transcriptome of embryonic and neonatal mouse cortex by high-throughput RNA sequencing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 31, p. 12741, 2009.

[165] UCSC Genome Browser. `http://genome.ucsc.edu`, Aug. 2011.

[166] National Center for Biotechnology Information (NCBI). `ftp://ftp.ncbi.nlm.nih.gov`, Aug. 2011.

[167] K. Pruitt, J. Harrow, R. Harte, *et al.*, "The consensus coding sequence (CCDS) project: Identifying a common protein-coding gene set for the human and mouse genomes," *Genome Research*, vol. 19, no. 8, 2009.

[168] H. Kikuta, M. Laplante, P. Navratilova, *et al.*, "Genomic regulatory blocks encompass multiple neighboring genes and maintain conserved synteny in vertebrates," *Genome Research*, vol. 17, no. 5, p. 545, 2007.

[169] P. Navratilova and T. Becker, "Genomic regulatory blocks in vertebrates and implications in human disease," *Briefings in Functional Genomics & Proteomics*, vol. 8, no. 4, p. 333, 2009.

[170] S. Arhondakis, K. Frousios, C. Iliopoulos, *et al.*, "Transcriptome map of mouse isochores," *BMC Genomics*, vol. 12, p. 511, 2011.

[171] K. Frousios, C. Iliopoulos, G. Tischler, *et al.*, "Transcriptome map of mouse isochores in embryonic and neonatal cortex," *Genomics*, vol. 101, no. 2, p. 120, 2013.

[172] R. Karchin, "Next generation tools for the annotation of human SNPs," *Briefings in Bioinformatics*, vol. 10, p. 35, 2008.

[173] S. Coassin, A. Brandstätter, and F. Kronenberg, "Lost in the space of bioinformatics tools: a constantly updated survival guide for genetic epidemiology. The GenEpi Toolbox," *Atherosclerosis*, vol. 209, p. 321, 2010.

[174] M. Cline and R. Karchin, "Using bioinformatics to predict the functional impact of SNVs," *Bioinformatics*, vol. 27, no. 4, p. 441, 2011.

[175] J. Thusberg, A. Olatubosun, and M. Vihinen, "Performance of mutation pathogenicity prediction methods on missense variants.," *Human Mutation*, vol. 32, no. 0, p. 1, 2011.

[176] P. Ng and S. Henikoff, "Predicting deleterious amino acid substitutions," *Genome Research*, vol. 11, p. 863, 2001.

[177] P. Kumar, S. Henikoff, and P. Ng, "Predicting the effect of coding non-synonynous variants on protein function using the SIFT algorithm.," *Nature Protocols*, vol. 4, no. 7, p. 1073, 2009.

[178] V. Ramensky, P. Bork, and S. Sunyaev, "Human non-synonymous SNPs: server and survey," *Nucleic Acids Research*, vol. 30, no. 17, p. 3894, 2002.

[179] P. Thomas, M. Campbell, A. Kejariwal, *et al.*, "PANTHER: A library of protein families and subfamilies indexed by function," *Genome Research*, vol. 13, p. 2129, 2003.

[180] P. Thomas, A. Kejariwal, M. Campbell, *et al.*, "PANTHER: A browsable database of gene products organized by biological function, using curated protein family and subfamily classification," *Nucleic Acids Research*, vol. 31, no. 1, p. 334, 2003.

[181] L. Conde, J. Vaquerizas, J. Santoyo, *et al.*, "PupaSNP Finder: a web tool for finding SNPs with putative effect at transcriptional level," *Nucleic Acids Research*, vol. 32, no. Web Server Issue, p. W242, 2004.

[182] L. Conde, J. Vaquerizas, C. Ferrer-Costa, *et al.*, "PupasView: a visual tool for selecting suitable SNPs, with putatve pathological effect in genes, for genotyping purposes," *Nucleic Acids Research*, vol. 33, no. Web Server issue, p. W501, 2005.

[183] J. Reumers, J. Schymkowitz, J. Ferkinghoff-Borg, *et al.*, "SNPeffect: a database mapping molecular phenotypic effects of human non-synonymous coding SNPs," *Nucleic Acids Research*, vol. 33, no. Database Issue, p. D527, 2005.

[184] G. De Baets, J. Van Durme, J. Reumers, *et al.*, "SNPeffect 4.0: on-line prediction of molecular and structural effects of protein coding variants," *Nucleic Acids Research*, vol. 40, no. Database Issue, p. D935, 2012.

[185] P. Yue and J. Moult, "Identification and anlysis of deleterious human SNPs," *Journal of Molecular Biology*, vol. 356, no. 5, p. 1263, 2006.

[186] P. Stenson, M. Mort, E. Ball, *et al.*, "The human gene mutation database: 2008 update," *Genome Med*, vol. 1, no. 1, p. 13, 2009.

[187] The UniProt Consortium, "Ongoing and future developments at the Universal Protein Resource," *Nucleic Acids Research*, vol. 39, no. Database issue, p. D214, 2011.

[188] P. Fujita, B. Rhead, A. Zweig, *et al.*, "The UCSC Genome Browser database: update 2011," *Nucleic Acids Research*, vol. 39, no. Database issue, p. D876, 2011.

[189] B. Giardine, C. Riemer, T. Hefferon, *et al.*, "PhenCode: connecting ENCODE data with mutations and phenotype," *Human Mutation*, vol. 28, p. 554, 2007.

[190] P. Baldi, S. Brunak, Y. Chauvin, *et al.*, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, p. 412, 2000.

[191] T. Joachims, "Making large-scale SVM learning practical.," in *Advances in Kernel Methods - Support Vector Learning* (B. Schoelkopf, C. Burges, and A. Smola, eds.), ch. 11, MIT-Press, 1999.

[192] K. Frousios, C. Iliopoulos, T. Schlitt, and M. Simpson, "Predicting the functional consequences of non–synonymous DNA sequence variants — evaluation of bioinformatics tools and development of a consensus strategy," *Genomics*, vol. 102, no. 4, p. 223, 2013.